

## **RESUMEN**

El contenido de este PFC supone el desarrollo de una aplicación gráfica en el entorno GNU/LINUX.

La aplicación debe representar en la pantalla gráficos procedentes de una fuente externa (un fichero, una tarjeta de adquisición de datos, otra aplicación, etc.)

Debe permitir las siguientes funcionalidades:

- Representar la señal uniendo puntos o marcándolos como símbolos.
- Debe gestionar de forma adecuada el zoom.
- Debe permitir exportar la gráfica a formatos SVG y PDF.
- Debe permitir mostrar varias señales superpuestas para poder compararlas.
- Opcionalmente debe permitir arrastrar y soltar (“drag and drop”).

## **ABSTRACT**

The content of the thesis is the development of a graphical application in GNU / Linux environment.

The application must render graphics on the screen from an external source (a file, a data acquisition card, other applications, etc..)

It must allow the following functionality:

- Joining dots represent the signal or marking them as symbols.
- Must properly manage the zoom.
- Allow export the graph to SVG and PDF formats.
- Allow display multiple signals superimposed for comparison.
- Optionally should allow drag and drop.

**PROYECTO FIN DE CARRERA**

# **"APLICACIÓN DE REPRESENTACIÓN GRÁFICA DE DATOS Y/O SEÑALES EN ENTORNO GNU/LINUX"**



**INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN**

**ESPECIALIDAD TELEMÁTICA**

**Autor: LORENZO SALAS PORTUGUÉS**

**Tutor: JOSÉ MANUEL PARDO MARTÍN**



## INDICE

<b>1. INTRODUCCIÓN .....</b>	<b>6</b>
<b>2. ENTORNO GNU/LINUX .....</b>	<b>7</b>
<b>2.1 UBUNTU .....</b>	<b>8</b>
<b>2.2 ANJUTA .....</b>	<b>9</b>
<b>2.3 GLADE .....</b>	<b>10</b>
<b>2.4 GTK .....</b>	<b>11</b>
<b>2.5 CAIRO .....</b>	<b>12</b>
<b>3. ANÁLISIS DE ESPECIFICACIONES DE LA APLICACIÓN .....</b>	<b>13</b>
<b>3.1 REQUISITOS FUNCIONALES .....</b>	<b>13</b>
<b>3.2 REQUISITOS NO FUNCIONALES .....</b>	<b>14</b>
<b>4. DISEÑO DE LA APLICACIÓN .....</b>	<b>15</b>
<b>4.1 MÉTODO UTILIZADO .....</b>	<b>15</b>
<b>4.2 PROGRAMACIÓN ORIENTADA A EVENTOS .....</b>	<b>16</b>
<b>4.3 PATRÓN DE DISEÑO: OBSERVER .....</b>	<b>17</b>
<b>4.4 DIAGRAMAS DE BLOQUE .....</b>	<b>18</b>
<b>4.5 DIAGRAMAS DE FLUJO .....</b>	<b>19</b>
a) Abrir la aplicación sin parámetros .....	19
b) Abrir la aplicación con parámetros .....	20
c) Evento abrir un fichero con la aplicación abierta .....	21
d) Evento guardar los datos en un fichero .....	22
e) Evento aumentar/disminuir valor de zoom eje X o Y pulsando el botón .....	23
f) Evento selección de señal, común a las 4 señales .....	24
g) Evento selección de rejilla .....	25
h) Evento selección dots .....	26
i) Evento selección color de fondo .....	27
j) Evento selección de un área a ampliar .....	28
k) Evento selección de señal para cálculo de máximos y mínimos. ....	29
l) Evento pulsar botón reset zoom. ....	30
m) Evento selección de color .....	31
<b>5. MANUAL DE USUARIO .....</b>	<b>33</b>
<b>5.1 INTERFAZ DE USUARIO .....</b>	<b>33</b>
<b>5.2 INICIO DE LA APLICACIÓN .....</b>	<b>35</b>
<b>5.3 FORMATO DE LOS FICHEROS DE DATOS .....</b>	<b>36</b>
<b>5.4 CARGAR DATOS .....</b>	<b>37</b>
<b>5.5 CAMBIAR EL COLOR DE UNA SEÑAL .....</b>	<b>39</b>
<b>5.6 CAMBIAR EL COLOR DE FONDO .....</b>	<b>41</b>
<b>5.7 DESACTIVAR LA REJILLA .....</b>	<b>42</b>
<b>5.8 MOSTRAR PUNTOS O SEGMENTOS .....</b>	<b>43</b>
<b>5.9 CÁLCULO DE MÁXIMOS Y MÍNIMOS .....</b>	<b>44</b>
<b>5.10 ZOOM .....</b>	<b>46</b>
<b>5.11 SELECCIÓN DE SEÑALES .....</b>	<b>48</b>
<b>5.12 EXPORTAR A PDF .....</b>	<b>50</b>
<b>5.13 EXPORTAR A SVG .....</b>	<b>51</b>
<b>5.14 EXPORTAR A PNG .....</b>	<b>52</b>



<b>6. CÓDIGO FUENTE .....</b>	<b>53</b>
<b>7. FUTURAS LÍNEAS DE TRABAJO .....</b>	<b>54</b>
<b>8. CONCLUSIONES .....</b>	<b>55</b>
<b>9. BIBLIOGRAFÍA .....</b>	<b>57</b>
<b>10. GLOSARIO.....</b>	<b>59</b>



## AGRADECIMIENTOS

A mi profesor José Manuel Pardo por ayudarme a realizar este proyecto con su apoyo, resolviendo todas las dudas que han ido surgiendo en el camino.

A Vanessa por aceptar mis “peculiaridades” y todo el tiempo que he invertido en la Universidad. Gracias por apoyarme en todo momento y no dejar que me rindiera.

A mis padres Lorenzo y Luisa, por educarme de la forma que lo han hecho y por apoyarme a superar todos los retos que me he propuesto. GRACIAS.

A mi hermano Rubén por la relación tan especial que tenemos y que nos ha vinculado desde que nacimos.

A mis hermanas María Luisa, Blanca Rosa y Esther por animarme en todo momento en mis estudios y hacerme sentir un hermano especial.

A mis amigos Agustín, David, Eva, Cristian, Ester y Sonia por esas noches de cenas y terracitas, por esas interminables partidas de dardos y todos los buenos momentos que hemos pasado juntos.

A todos los amigos y compañeros del trabajo que me han apoyado y facilitado el poder terminar mis estudios con mis horarios “especiales” y mis desapariciones en épocas de exámenes.

A todos los amigos y compañeros de la Universidad que me han ayudado y apoyado, por todos esos momentos pasados en la biblioteca y en la cafetería, ayudándonos y aprendiendo a trabajar en equipo.

En definitiva a todas las personas que se han cruzado en mi camino (profesores, compañeros, vecinos...) y que han ido dejando su pequeña aportación y me han hecho ir creciendo tanto en el terreno personal como profesional, y que sin duda me seguiré encontrando en el futuro.



Para terminar deseo incluir una frase que escribió el matemático Pedro Puig Adam (1900-1969) y que siempre la he tenido presente:

***"Tended a ser un poco aprendices de todo para vuestro bien y, al menos,  
especialistas en algo para bien de los demás."***



## 1. INTRODUCCIÓN

El principal objetivo de este Proyecto es el desarrollo de una aplicación GNU/LINUX que represente gráficamente señales en dos dimensiones.

En la actualidad existen varias aplicaciones que muestran gráficamente señales en entorno GNU/LINUX, pero básicamente sólo representan las señales, sin permitir modificarlas ni ampliarlas de una forma cómoda y práctica para el usuario.

La funcionalidad principal de la aplicación será la de gestionar el ZOOM de forma dinámica para que se adapte a las necesidades del usuario, permitiendo el uso del ratón para ampliar zonas determinadas.

Además se incluirán otras funcionalidades como cálculo de máximos y mínimos y la posibilidad de exportar las gráficas a otros formatos.

Se ha elegido UBUNTU como sistema operativo por ser una de las distribuciones de Linux más extendida en la actualidad.

Para el desarrollo de la aplicación se ha utilizado ANJUTA como IDE (entorno de desarrollo integrado) y GLADE-3 para el diseño de la interfaz gráfica.

Se ha utilizado C como lenguaje de programación para esta aplicación, y se han utilizado las librerías GTK para la interfaz y CAIRO para dibujar los gráficos.

El presente documento se organiza en varios capítulos. El primero es la presente introducción. En el segundo se muestra el entorno que se ha utilizado para la implementación de esta aplicación, mostrando una a una las herramientas que han sido necesarias. En el tercero se realiza un análisis de los requisitos que se han contemplado para el diseño. El cuarto explica el diseño completo de la aplicación. El quinto es un breve manual de usuario que explica cómo se utiliza la aplicación. El sexto es el código fuente de la aplicación. El séptimo son las futuras líneas de trabajo, con las que se podría continuar desarrollando este proyecto. Y por último las conclusiones, bibliografía y glosario completan la memoria de este proyecto.





## 2. ENTORNO GNU/LINUX<sup>1</sup>

El entorno GNU/LINUX es un sistema operativo que utiliza como núcleo el kernel de LINUX. El nombre GNU (que se pronuncia «ñu» en inglés) significa “GNU No es UNIX”.

El proyecto GNU comenzó en 1984 desarrollado por Richard Stallman. El principal objetivo era desarrollar un sistema basado en UNIX pero libre. La combinación de GNU y LINUX como sistema operativo, se usa por un gran número de usuarios de todos los niveles y se caracteriza por la robustez y estabilidad. Popularmente es conocido como LINUX aunque la denominación correcta sería GNU/LINUX.

El proyecto GNU/LINUX está vinculado a la filosofía de Software Libre. El Software Libre se refiere a la libertad de expresión y no implica que no tenga un coste. Concretamente se refiere a los **cuatro tipos de libertades** para los usuarios de software:

1. Libertad de usar el programa con cualquier propósito.
2. Libertad de estudiar el funcionamiento del programa y adaptarlo a las necesidades. Esto implica el acceso al código fuente.
3. Libertad de distribuir copias con las que poder ayudar a otros usuarios.
4. Libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad de usuarios se beneficie.

---

<sup>1</sup> Fuente: <http://www.gnu.org/home.es.html> y [http://geekubuntu.blogspot.com.es/2012\\_04\\_01\\_archive.html](http://geekubuntu.blogspot.com.es/2012_04_01_archive.html)



## 2.1 UBUNTU<sup>2</sup>

UBUNTU es un sistema operativo basado en Debian GNU/Linux. Se caracteriza por la libertad de uso y por su fácil instalación. Aunque está principalmente enfocado para ordenadores de escritorio también existen versiones para servidores.

Ofrece actualizaciones semestrales, en Abril y Octubre, y las versiones se nombran usando las cifras del año, un punto y la cifra del mes (por ejemplo la última versión sería la 12.4).

UBUNTU está patrocinada por Canonical Ltd. es una empresa fundada y financiada por el empresario sudafricano Mark Shuttleworth.

Los siguientes principios son la base de la filosofía UBUNTU:

- UBUNTU siempre será gratuito y estará disponible para cualquier usuario incluso para uso profesional.
- UBUNTU usa las mejores herramientas de traducción y accesibilidad para que pueda ser usado por el mayor número de usuarios posible.
- UBUNTU publica una nueva versión cada seis meses de manera regular y predecible.
- UBUNTU tiene un compromiso total con los principios de desarrollo software de código abierto, de forma que los usuarios pueden usarlo, mejorarlo y compartirlo.

UBUNTU incluye un repositorio de aplicaciones llamado SYNAPTIC con el que se pueden descargar e instalar aplicaciones de todo tipo de forma gratuita. El modo de funcionamiento es muy similar a PLAY STORE de dispositivos ANDROID o a APP STORE de dispositivos APPLE.

---

<sup>2</sup> Fuentes: <http://www.guia-ubuntu.org/index.php?title=Ubuntu> y <http://tecnoubuntu.wordpress.com/%C2%BFde-que-ma-hablas/>

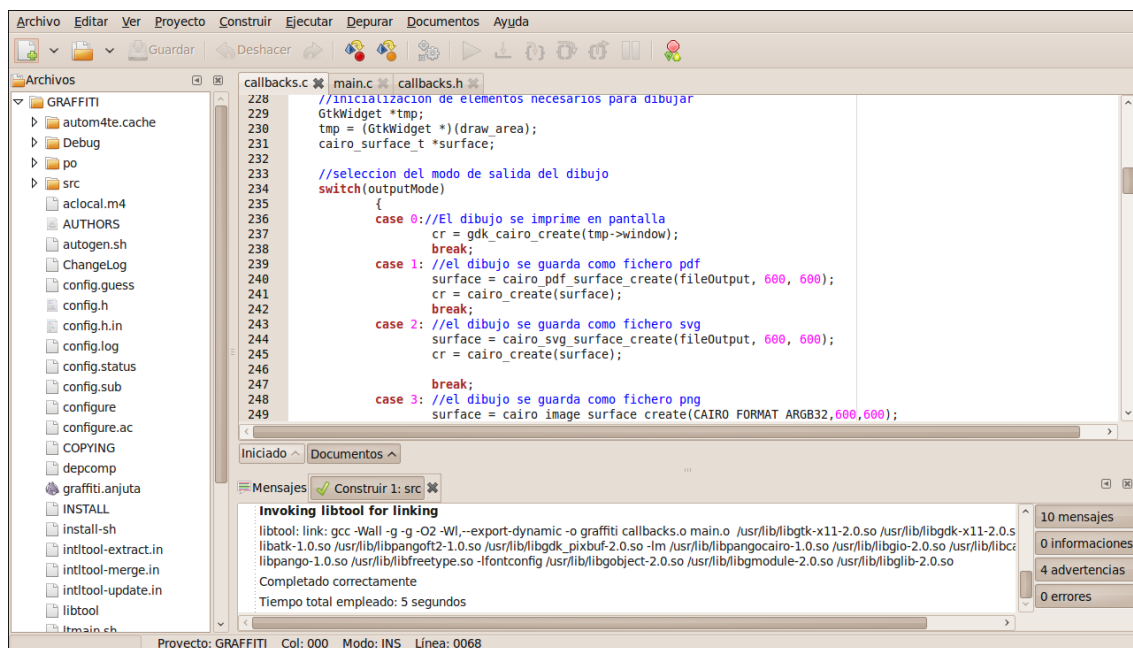


## 2.2 ANJUTA<sup>3</sup>

ANJUTA es un entorno de desarrollo integrado (IDE). Es gratuito y de código abierto. Permite programar en distintos lenguajes como C, C++, Java, etc. Principalmente se usa para programar con GTK+ en entornos de escritorio GNOME.

Es un programa compuesto por un conjunto de herramientas para el programador. Consta de un editor de código, un compilador, un depurador, un gestor de proyectos y está perfectamente integrado con GLADE (un constructor de interfaz gráfica que se explica en el apartado 2.3).

La siguiente captura muestra el aspecto de la interfaz de ANJUTA con el código de la aplicación del presente Proyecto abierta.



<sup>3</sup> Fuente: <http://anjuta.uptodown.com/ubuntu>



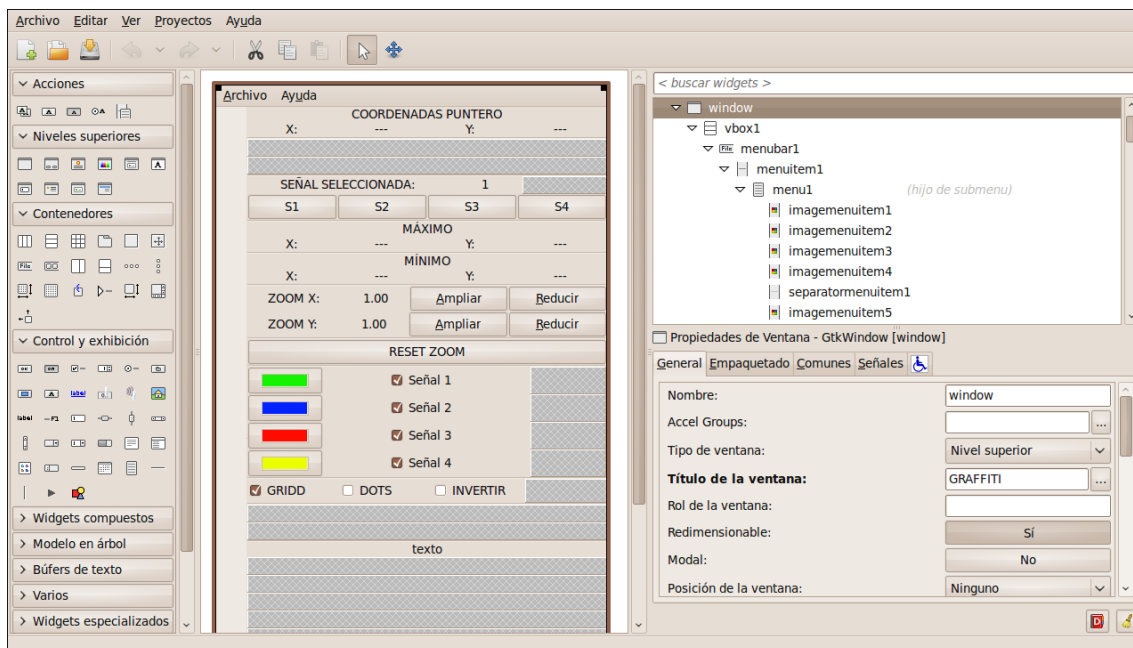
## 2.3 GLADE<sup>4</sup>

GLADE es una herramienta de diseño y desarrollo de interfaz gráfica de usuario. Se caracteriza por su facilidad de uso al ser visual. Además es gratuita con licencia GNU.

El diseño principal se hace arrastrando y soltando elementos a la ventana. A continuación se configuran los elementos permitiendo elegir desde el nombre hasta los eventos que generan o soportan. Como resultado se crea un fichero XML con la base de datos de todos los componentes que es leído en tiempo de ejecución de la aplicación. Esto permite modificar la interfaz en cualquier momento del diseño e implementación de una aplicación, de manera independiente, lo que facilita la posibilidad de hacer cambios.

Es independiente del lenguaje de programación y compatible con ANJUTA y es básico para desarrollar aplicaciones GTK.

A continuación se muestra una captura de GLADE con la interfaz de la aplicación abierta.



<sup>4</sup> Fuente: <http://glade.uptodown.com/ubuntu>



## 2.4 GTK<sup>5</sup>

GTK (Gimp ToolKit) es un conjunto de bibliotecas multiplataforma. Es una poderosa herramienta para modelar y crear interfaces gráficas y principalmente se utilizada en entornos GNOME.

Sus componentes principales son:

- **GDK** (Gimp Drawing Kit): Se encarga de dibujar (polígonos, líneas), cargar y manipular imágenes y cualquier elemento gráfico.
- **GLIB** Conjunto de bibliotecas que conforman el corazón de GTK y GNOME. Principalmente se encarga de todo el manejo de estructuras de datos, portabilidad, el manejo del sistema de objetos, hilos y todo lo que corresponde al sistema base de GTK.
- **ATK** (Accessibility Toolkit): Conjunto de bibliotecas que permiten que GTK mantenga todas las opciones de accesibilidad.
- **Pango**: Se encarga del manejo de fuentes e internacionalización.

Se define con un modelo gobernado por eventos. El modelo está constituido por un despachador principal, el cual casi todo el tiempo está en espera de eventos. Cuando ocurre un evento es insertado en una cola en la cual cada uno de acuerdo a su turno emite una señal de suceso y el programador decide qué función atiende a cada evento específico.

---

<sup>5</sup> Fuente: <http://underc0de.org/foro/perl/programacion-gui-perl-gtk/>



## 2.5 CAIRO<sup>6</sup>

CAIRO es una librería para dibujar gráficos en dos dimensiones que soportan múltiples sistemas como X Window System, Quartz, Win32, PostScript, etc. Además permite exportar los gráficos a otros formatos como PDF, SVG o PNG.

CAIRO está diseñado para producir una representación consistente en todos los dispositivos de salida y aprovecha la aceleración de hardware de la pantalla cuando está disponible.

CAIRO proporciona operaciones similares a los operadores de dibujo de PostScript y PDF, incluyendo suavizado y llenando splines cúbicos, transformación y composición de imágenes translúcidas y representación de texto. En todas las operaciones de dibujo se puede realizar cualquier transformación como por ejemplo cambio de escala, rotación, corte, etc.

CAIRO se implementa como una biblioteca escrita en el lenguaje de programación C, pero los enlaces están disponibles para varios lenguajes de programación diferentes.

CAIRO es un software gratuito y está disponible para ser redistribuido o modificado bajo los términos de la GNU.

---

<sup>6</sup> Fuente: <http://www.cairographics.org/>  
y <http://developer.gnome.org/platform-overview/stable/cairo.html.es>



### **3. ANÁLISIS DE ESPECIFICACIONES DE LA APLICACIÓN**

Las especificaciones que la aplicación debe cumplir y a partir de las cuales se realiza el diseño y desarrollo de la aplicación se dividen en dos partes: requisitos funcionales y no funcionales.

#### **3.1 REQUISITOS FUNCIONALES**

Los requisitos funcionales son los servicios esperados por la aplicación y que no imponen estrategias de diseño. Se enumeran a continuación:

- Mostrar simultáneamente varias señales de 2 dimensiones.
- Cada señal tendrá un color distinto por defecto, el cual podrá ser modificado.
- La lectura de los datos será a través de un fichero de texto y la entrada estándar.
- El zoom funcionará de forma dinámica, se podrá arrastrar y soltar sobre la ventana con el ratón y mediante botones.
- La visualización individual de cada señal será seleccionable.
- Los datos de las señales seleccionadas se podrán visualizar como una nube de puntos o uniendo cada punto con líneas.
- La aplicación calcula los valores de máximos y mínimos de cada una de las señales.
- La aplicación calcula automáticamente la escala para ajustar la señal a la ventana de visualización.
- La rejilla (GRIDD) se podrá activar y desactivar.



### 3.2 REQUISITOS NO FUNCIONALES

Los requisitos no funcionales son las obligaciones impuestas a la aplicación que imponen restricciones para el diseño. Se enumeran a continuación:

- Será estable ante entradas no válidas de datos.
- Mostrará simultáneamente hasta 4 señales distintas superpuestas.
- El fichero de texto del cual se leerán los datos tendrá el siguiente formato: en la primera línea dos números N y M. N indica el número de filas (muestras) que hay en el fichero y M el número de señales y en las siguientes líneas los valores de cada uno de los valores capturados en este orden: X Y1 Y2 Y3 Y4.
- La lectura de datos en la entrada estándar usará el mismo formato que el fichero de texto.
- El zoom se representará con dos decimales.
- La memoria se gestionará de forma dinámica para consumir los mínimos recursos del sistema, se usarán listas enlazadas y asignación dinámica de memoria en lugar de arrays.
- Permitirá exportar la imagen que se esté visualizando a formatos PDF, SVG y PNG.





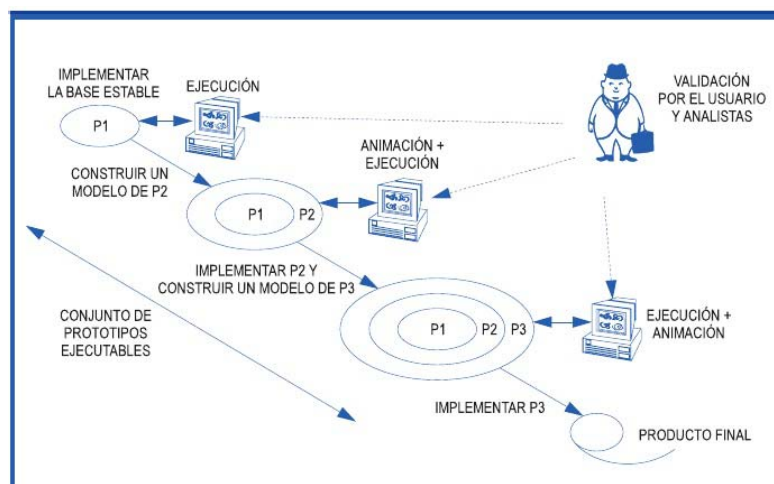
## 4. DISEÑO DE LA APLICACIÓN

### 4.1 MÉTODO UTILIZADO<sup>7</sup>

Para la presente aplicación se ha utilizado el método de diseño de prototipado incremental por ser el que mejor se adaptaba a las necesidades. Esto ha permitido implementar una aplicación ejecutable desde el principio del desarrollo con la que evaluar el cumplimiento de los requisitos. También ha permitido añadir y eliminar pequeños detalles que no se contemplaron inicialmente.

Al tratarse de una aplicación gráfica se le ha dado mucha importancia a la apariencia de la interfaz y desde el principio se ha podido modificar su aspecto externo según las necesidades de cada fase del desarrollo.

Las principales ventajas del uso de este método son aumentar la visibilidad en el desarrollo de la aplicación, permitiendo la experimentación con prototipos ejecutables intermedios y los modelos realizados siguen siendo empleados en el siguiente prototipo o se convierten en un modelo con un nivel de detalle mayor.



<sup>7</sup> Fuente <http://www.slideshare.net/jpbthames/modelos-de-desarrollo-rpido-de-software>



## 4.2 PROGRAMACIÓN ORIENTADA A EVENTOS<sup>8</sup>

La programación orientada a eventos es un paradigma de la programación, en el que el funcionamiento de un programa está definido por los eventos que sucedan durante la ejecución de la aplicación. Estos eventos pueden ser generados por el usuario o de forma interna por la propia aplicación.

Mientras que en la programación secuencial o estructurada es el programador el que dirige el flujo de ejecución de la aplicación, en la programación orientada a eventos son los propios eventos los que dirigen el flujo de ejecución del programa.

Cuando se desarrolla una aplicación orientada a eventos se deben definir qué eventos se manejan y qué código se ejecutará para cada uno. Los eventos dependerán del lenguaje de programación utilizado y del sistema operativo.

Al comenzar la ejecución de un programa orientado a eventos, después de un código inicial, se queda bloqueado a la espera de algún evento. Cuando sucede algún evento esperado, el programa ejecutará el código que corresponda a dicho evento.

---

<sup>8</sup> Fuente: <http://programarjava.wordpress.com/2011/12/13/programacion-orientada-a-eventos/>



### 4.3 PATRÓN DE DISEÑO: OBSERVER<sup>9</sup>

De los diferentes patrones de diseño que se pueden encontrar en numerosa bibliografía y páginas Web, se ha elegido el patrón Observer por ser el que mejor se adapta al diseño de la aplicación del presente Proyecto. Las principales razones son que se trata de un programa orientado a eventos y de una aplicación gráfica.

El patrón Observer se define por un manejador y despachador de eventos en el que uno o varios observadores se suscriben a un sujeto, el cual les notifica cuándo un evento se ha producido, para que los observadores puedan realizar una determinada operación.

Para el presente Proyecto toda la problemática de atender a los eventos se simplifica ampliamente por el uso de GLADE y GTK. La gestión de los eventos se hace así prácticamente transparente para el desarrollador, haciendo posible concentrar el esfuerzo en la programación de las rutinas de atención de cada evento.

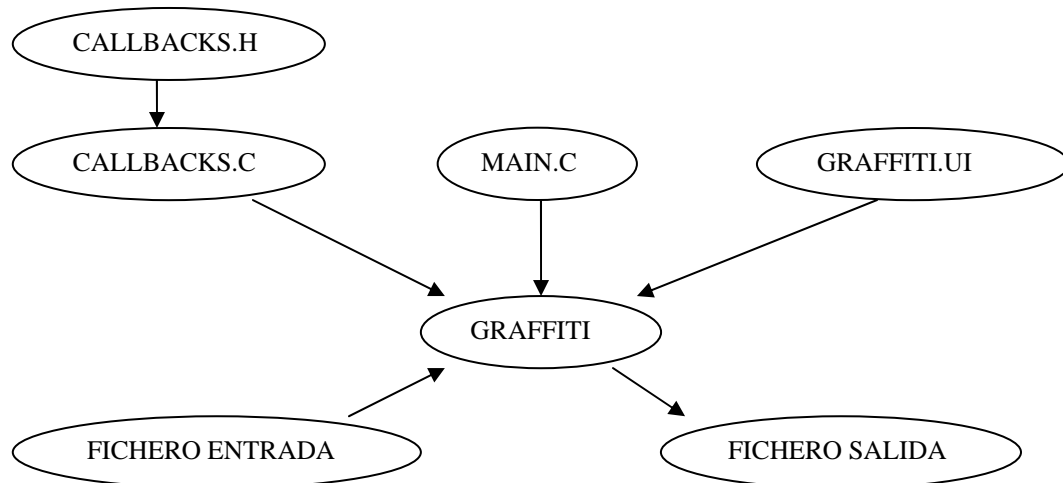
---

<sup>9</sup> Fuente: <http://arturoweb.wordpress.com/2010/03/08/patron-de-diseno-observer-en-php/#more-160>  
y <http://www.yaxche-soft.com/en/node/21>



#### 4.4 DIAGRAMAS DE BLOQUE

El diseño de la aplicación se explica con el siguiente diagrama:



**MAIN.C** Fichero donde reside el programa principal, enlaza las señales de los eventos y carga la base de datos de la interfaz gráfica.

**CALLBACKS.C** Fichero donde están todas las funciones de atención a eventos de la interfaz gráfica. Además, están algunas funciones internas y la definición de todas las variables y estructuras usadas.

**CALLBACKS.H** Fichero que almacena las cabeceras de **CALLBACKS.C** para que los nombres de las funciones y los parámetros sean conocidos por **MAIN.C**

**GRAFFITI.UI** Fichero creado con GLADE en el que se modela la base de datos de la interfaz gráfica. Se carga en tiempo de ejecución de la aplicación.

**FICHEROS DE ENTRADA** Fichero de texto plano, que almacena los datos que se quieren representar y que cumple con el formato descrito en los requisitos de la aplicación.

**FICHEROS DE SALIDA** Ficheros de tipo PDF, SVG o PNG en los que se almacenan los resultados de las modificaciones que se le hagan a la señal de entrada.



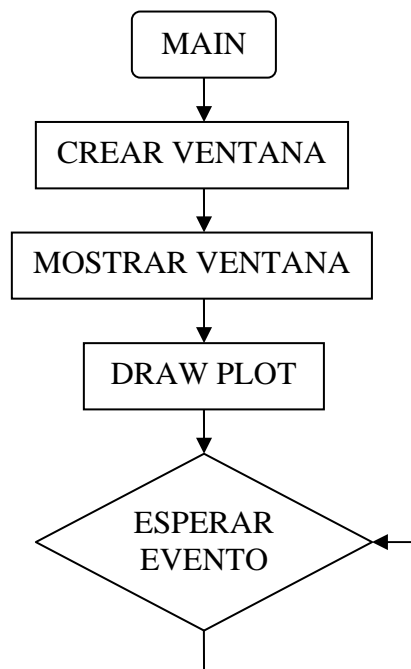
#### 4.5 DIAGRAMAS DE FLUJO

La forma más adecuada de describir el diseño y funcionamiento de la presente aplicación es mediante una serie de diagramas de flujo. Cada uno describe distintas situaciones y eventos que se pueden dar en la ejecución de la aplicación.

Los diagramas a) y b) parten de la aplicación cerrada.

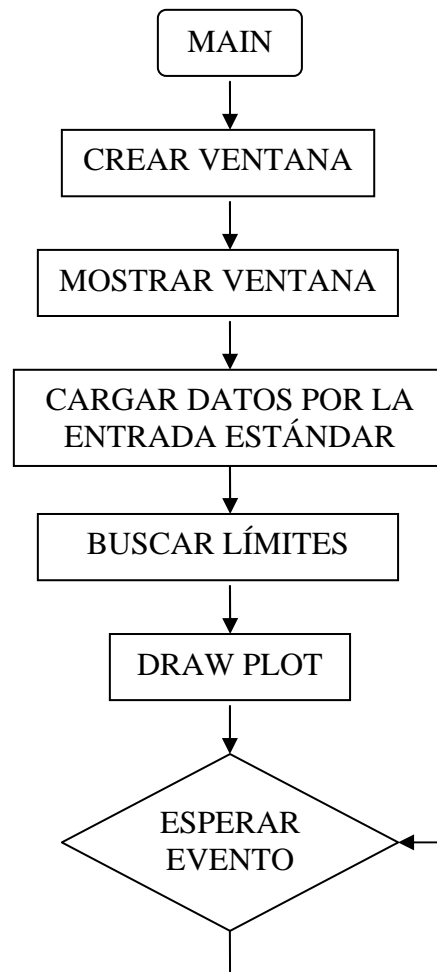
En el resto de diagramas la aplicación está abierta y a la espera de nuevos eventos.

##### a) Abrir la aplicación sin parámetros



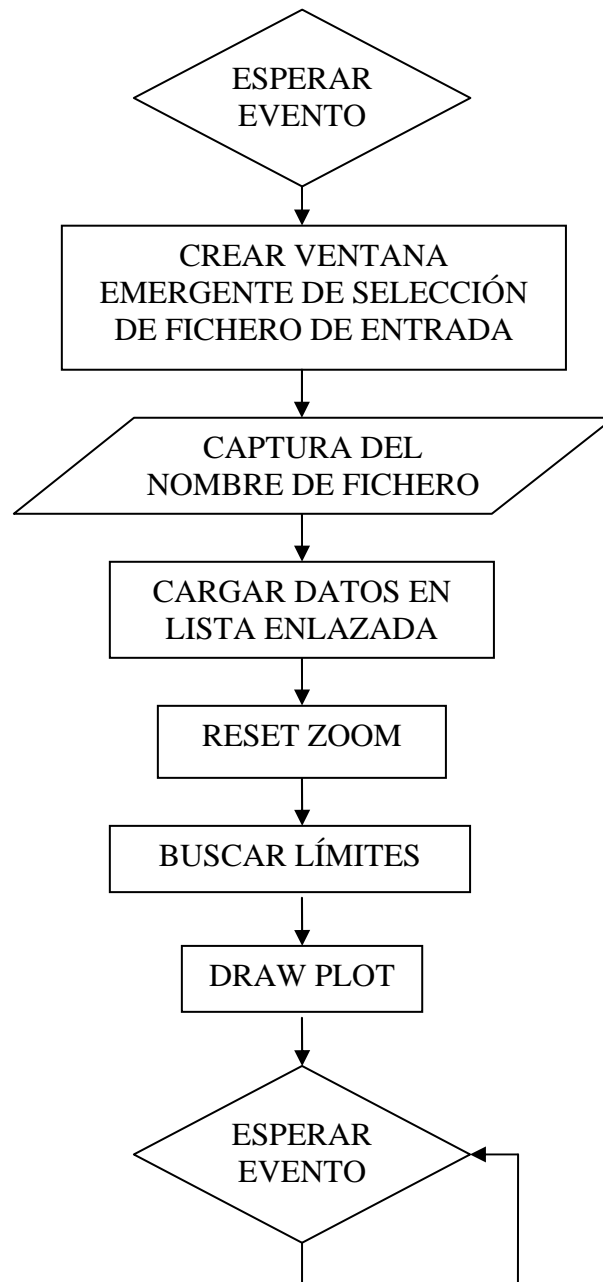


**b) Abrir la aplicación con parámetros**



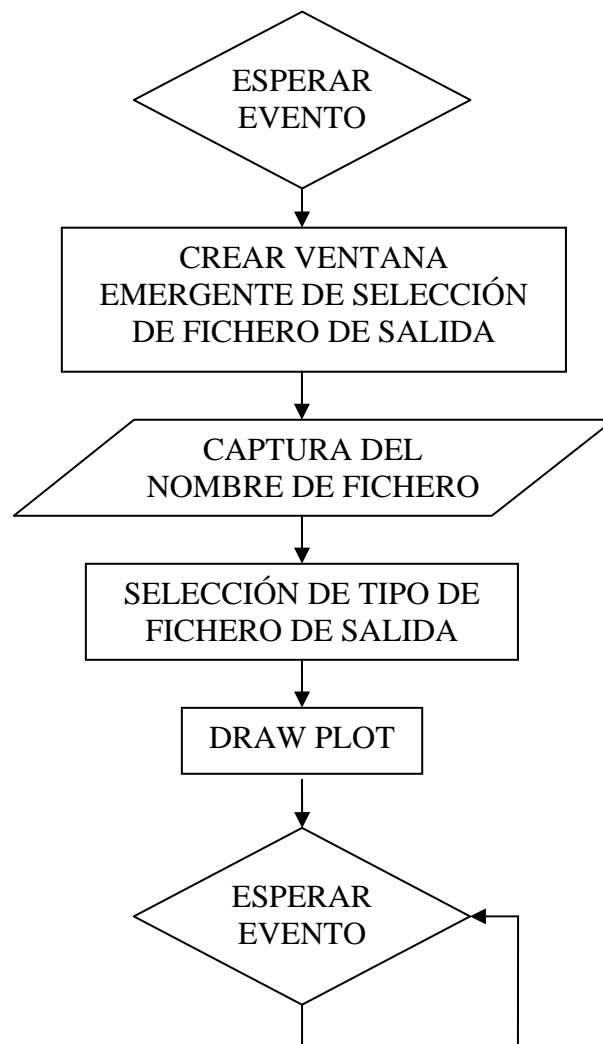


**c) Evento abrir un fichero con la aplicación abierta**





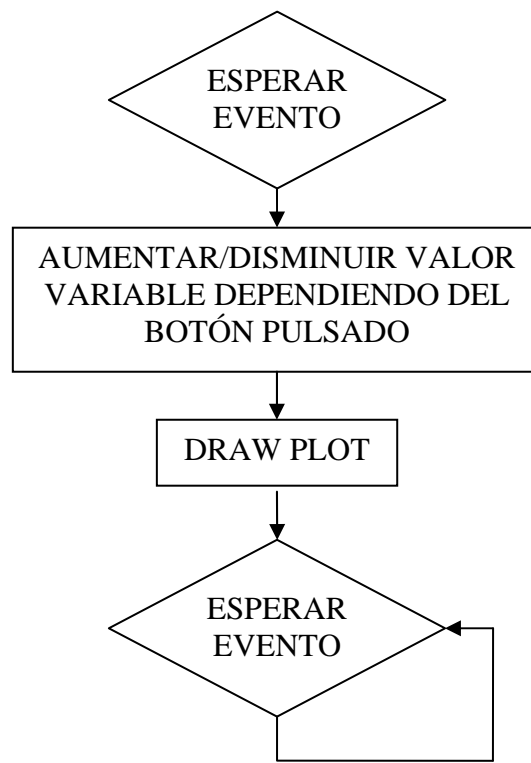
**d) Evento guardar los datos en un fichero**





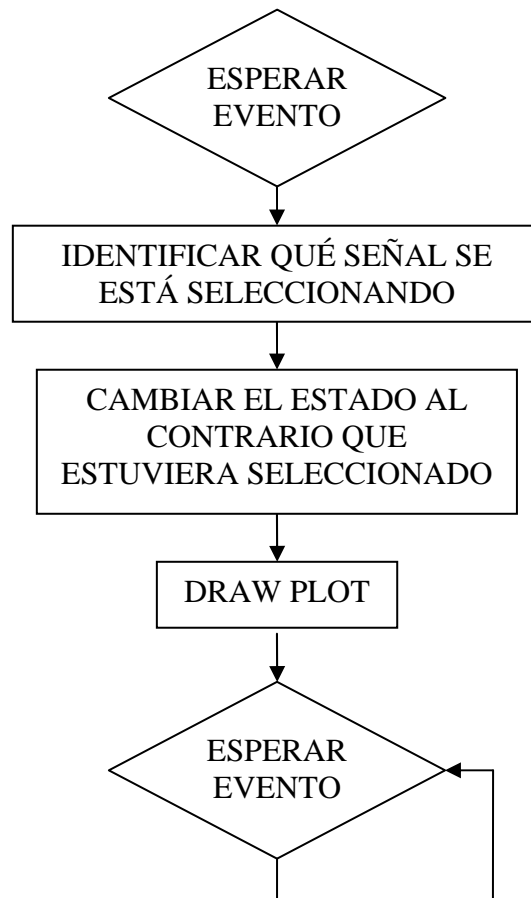


**e) Evento aumentar/disminuir valor de zoom eje X o Y pulsando el botón**



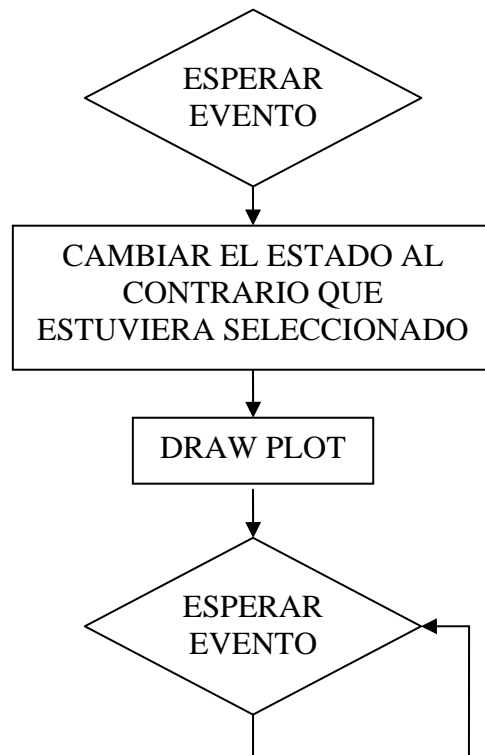


**f) Evento selección de señal, común a las 4 señales**



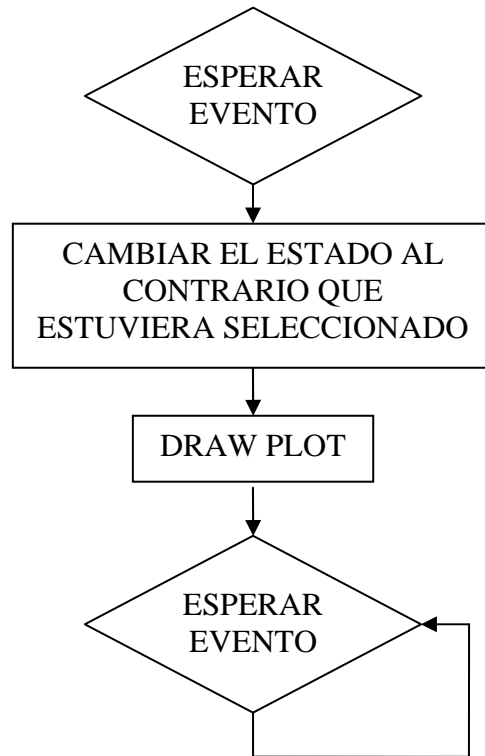


**g) Evento selección de rejilla**



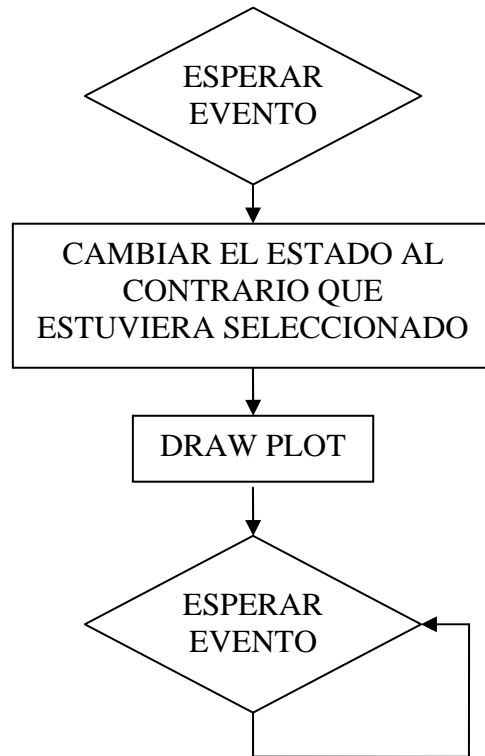


**h) Evento selección dots**



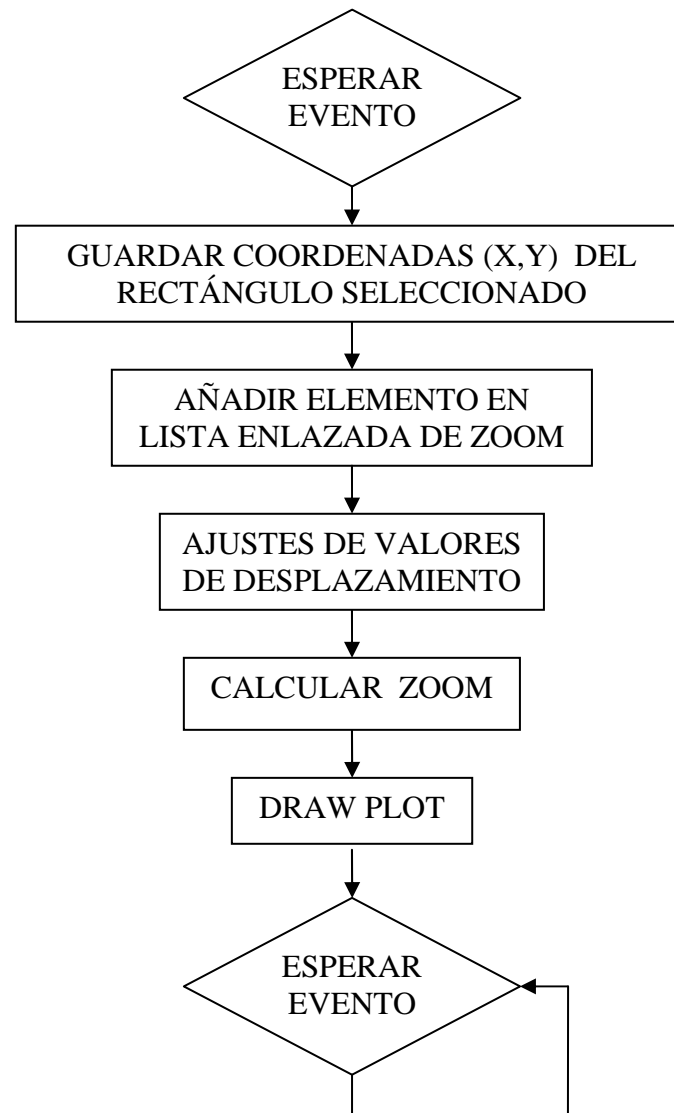


**i) Evento selección color de fondo**



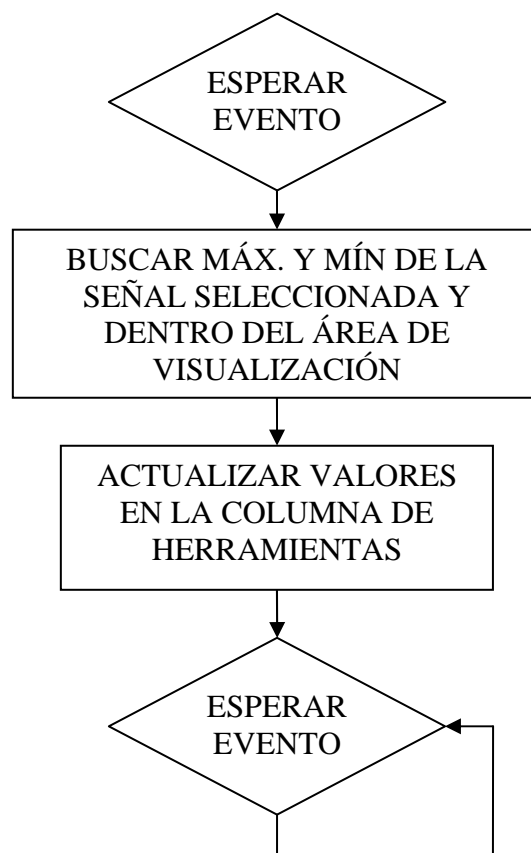


**j) Evento selección de un área a ampliar**



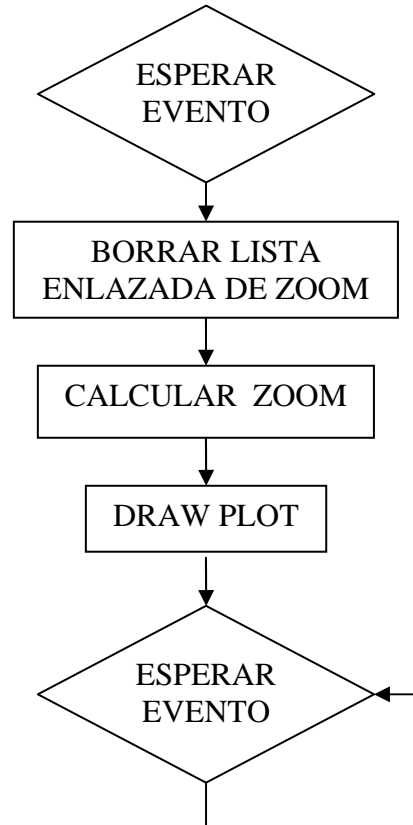


**k) Evento selección de señal para cálculo de máximos y mínimos.**





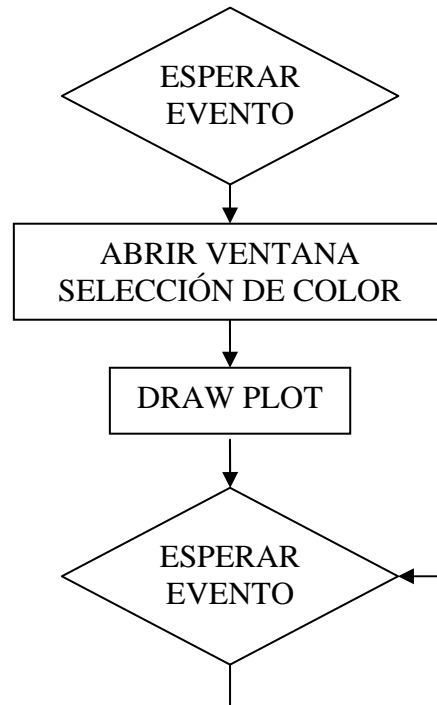
**1) Evento pulsar botón reset zoom.**







**m) Evento selección de color**





#### **n) Función DRAW PLOT**

Es la función responsable de actualizar la ventana de visualización cada vez que hay algún cambio producido por algún evento o cuando se abre la aplicación.

Como se trata de una función muy extensa (más de 300 líneas de código) en la que se realizan muchas operaciones, en el siguiente listado se muestra un resumen de las operaciones que se llevan a cabo.

- Inicializar la lista enlazada de ZOOM si no se ha hecho antes.
- Inicializar las variables necesarias para dibujar.
- Seleccionar el modo de salida del dibujo (pantalla, PDF, SVG o PNG).
- Dibujar el color de fondo según lo que se haya seleccionado.
- Dibujar rejilla si está seleccionada.
- Dibujar escalas.
- Capturar los colores de cada señal.
- Dibujar las señales seleccionadas y en forma de puntos o segmentos.
- Eliminar segmentos de señal que no tienen que ser mostrados.
- Dibujar leyendas y valores de las escalas.
- Imprimir el dibujo según el modo de salida seleccionado.
- Buscar máximos y mínimos.

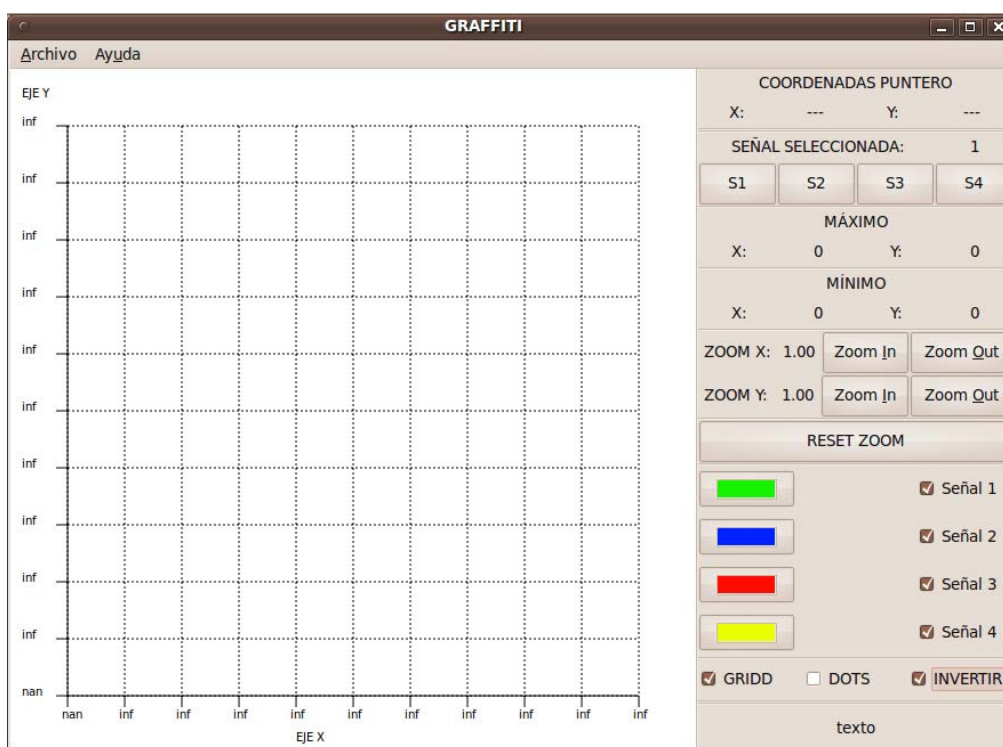


## 5. MANUAL DE USUARIO

La aplicación GRAFFITI es una aplicación que permite representar gráficamente hasta 4 señales de dos dimensiones simultáneamente. A continuación se explican con detalle cómo se ejecuta y cómo se utiliza todas las funcionalidades de la aplicación.

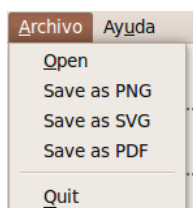
### 5.1 INTERFAZ DE USUARIO

La interfaz de usuario de GRAFFITI es muy sencilla e intuitiva para facilitar el uso de la aplicación a cualquier usuario, independientemente de sus conocimientos y experiencia. A continuación se presenta una captura de la interfaz de usuario:



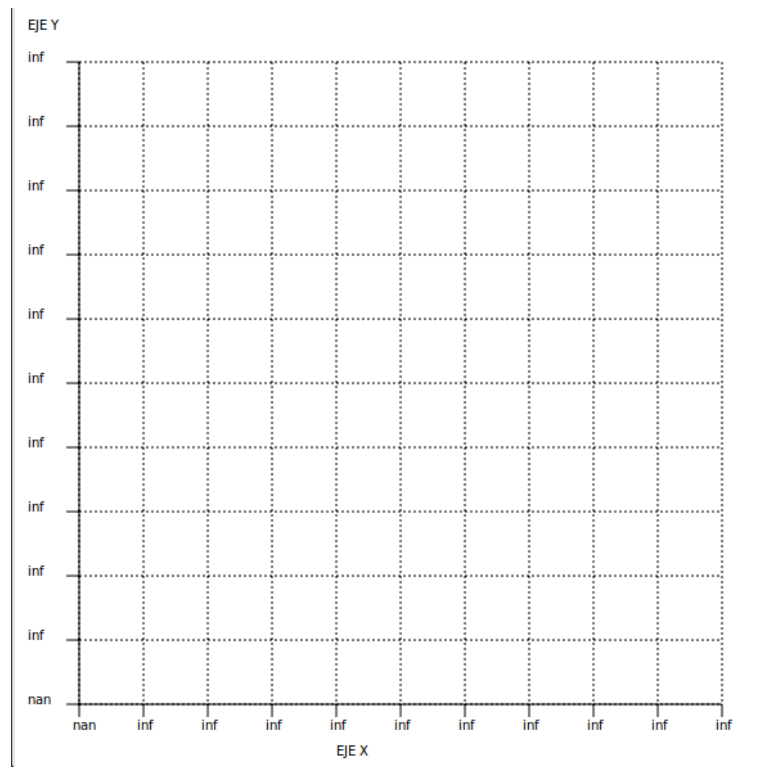
La interfaz se divide en tres partes:

- 1) Una barra de menú desplegable en la que encontramos los menús de abrir fichero, guardar en distintos formatos y salir.





- 2) Un cuadrado de dibujo en el que se muestran tanto las señales como los valores de la rejilla. Esta zona es sensible al ratón de modo que se puede ampliar la imagen arrastrando y soltando para seleccionar un área.



- 3) Una columna de herramientas con todos los botones necesarios para modificar la representación (colores, ZOOM, selección de señales, etc.) y también unos cuadros en los que muestra los máximos y mínimos de la señal seleccionada.

COORDENADAS PUNTERO	
X: ---	Y: ---
SEÑAL SELECCIONADA: 1	
S1	S2 S3 S4
MÁXIMO	
X: 0	Y: 0
MÍNIMO	
X: 0	Y: 0
ZOOM X: 1.00	Zoom In Zoom Out
ZOOM Y: 1.00	Zoom In Zoom Out
RESET ZOOM	
	<input checked="" type="checkbox"/> Señal 1
	<input checked="" type="checkbox"/> Señal 2
	<input checked="" type="checkbox"/> Señal 3
	<input checked="" type="checkbox"/> Señal 4
<input checked="" type="checkbox"/> GRIDD	<input type="checkbox"/> DOTS <input checked="" type="checkbox"/> INVERTIR
texto	



## 5.2 INICIO DE LA APLICACIÓN

La aplicación se puede ejecutar de tres formas:

- 1) Desde el terminal, ejecutando el nombre de la aplicación con o sin parámetros.

```
Archivo  Editar  Ver  Terminal  Ayuda
administrator@ubuntu:~/work/GRAFFITI/src$
administrator@ubuntu:~/work/GRAFFITI/src$
administrator@ubuntu:~/work/GRAFFITI/src$
administrator@ubuntu:~/work/GRAFFITI/src$
administrator@ubuntu:~/work/GRAFFITI/src$
administrator@ubuntu:~/work/GRAFFITI/src$
administrator@ubuntu:~/work/GRAFFITI/src$
administrator@ubuntu:~/work/GRAFFITI/src$ /home/administrator/work/GRAFFITI/src/graffiti
```

- 2) Desde el explorador de archivos, haciendo clic en el icono ejecutable.



- 3) Desde otra aplicación que desencadene la ejecución.



### 5.3 FORMATO DE LOS FICHEROS DE DATOS

El formato de los datos utilizados por GRAFFITI es el siguiente:

<b>M</b>	<b>N</b>			
<b>X</b>	<b>Y1</b>	<b>Y2</b>	<b>Y3</b>	<b>Y4</b>
<b>X'</b>	<b>Y1'</b>	<b>Y2'</b>	<b>Y3'</b>	<b>Y4'</b>
<b>X''</b>	<b>Y1''</b>	<b>Y2''</b>	<b>Y3''</b>	<b>Y4''</b>
<b>X'''</b>	<b>Y1'''</b>	<b>Y2'''</b>	<b>Y3'''</b>	<b>Y4'''</b>

**M:** número de muestras del fichero (número de filas sin incluir la primera).

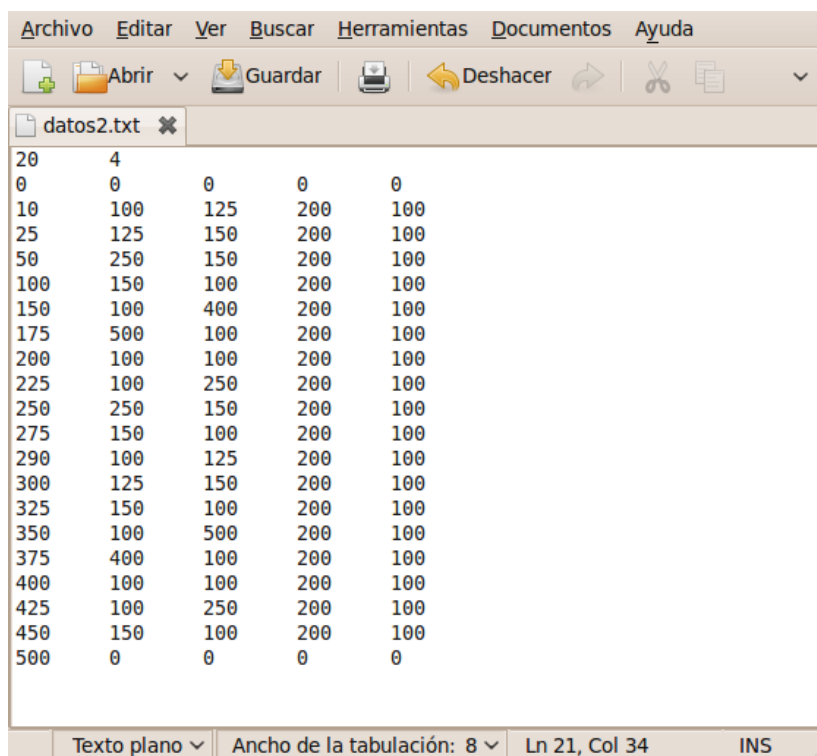
**N:** número de señales de las que se ha tomado muestras. Varía entre 1 y 4.

**X, X', X''.....:** valores de las coordenadas X.

**Y1, Y1', Y1''.....:** valores de las coordenadas Y1.

Los valores de las siguientes columnas tienen el mismo significado para las otras señales.

A continuación se muestra un ejemplo del aspecto que tendría el contenido de un fichero de datos válido:



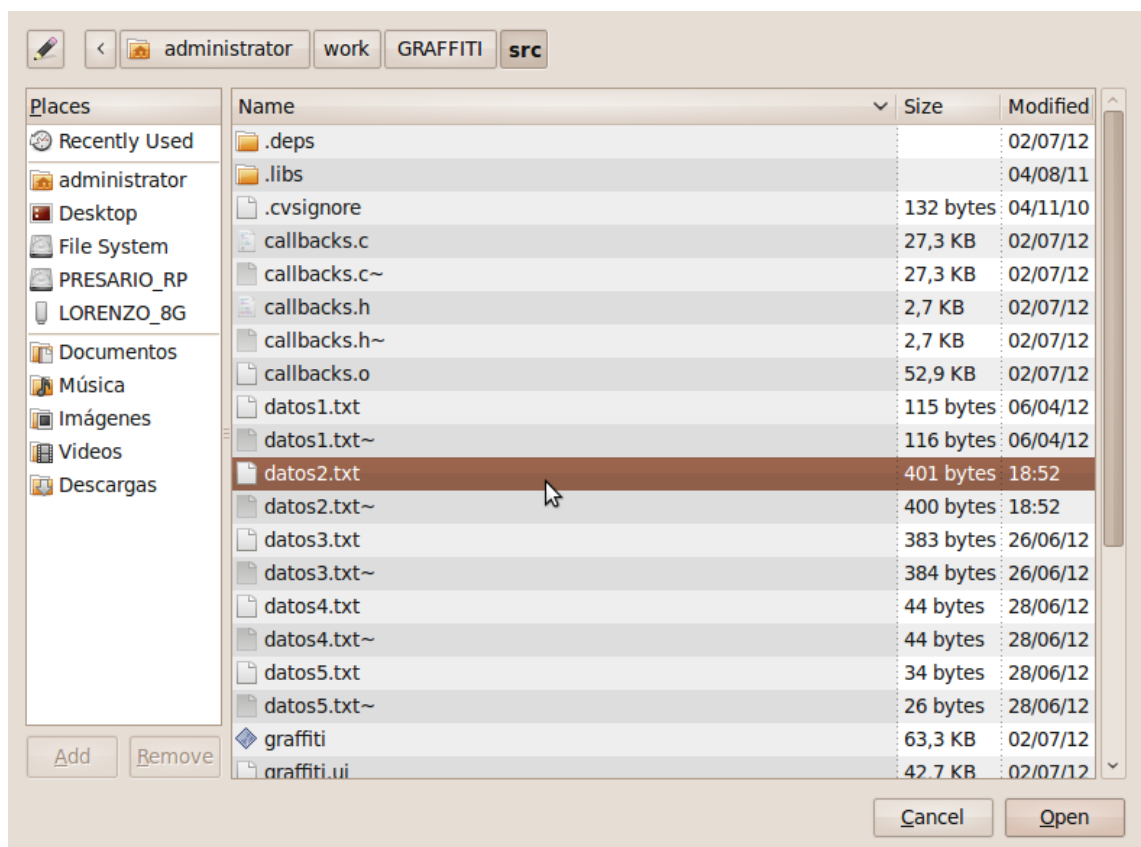
20	4			
0	0	0	0	0
10	100	125	200	100
25	125	150	200	100
50	250	150	200	100
100	150	100	200	100
150	100	400	200	100
175	500	100	200	100
200	100	100	200	100
225	100	250	200	100
250	250	150	200	100
275	150	100	200	100
290	100	125	200	100
300	125	150	200	100
325	150	100	200	100
350	100	500	200	100
375	400	100	200	100
400	100	100	200	100
425	100	250	200	100
450	150	100	200	100
500	0	0	0	0



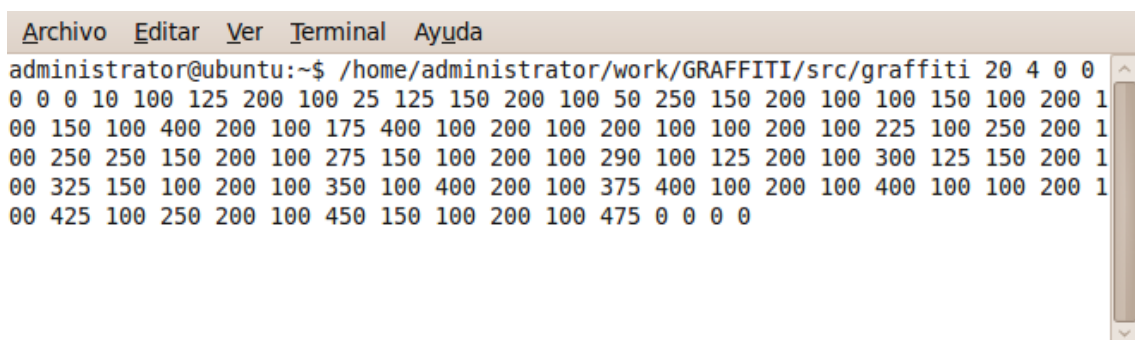
## 5.4 CARGAR DATOS

La carga de datos se puede realizar de dos formas:

- 1) Desde el menú abrir y seleccionando el fichero en el que se encuentra los datos que se quieren representar.

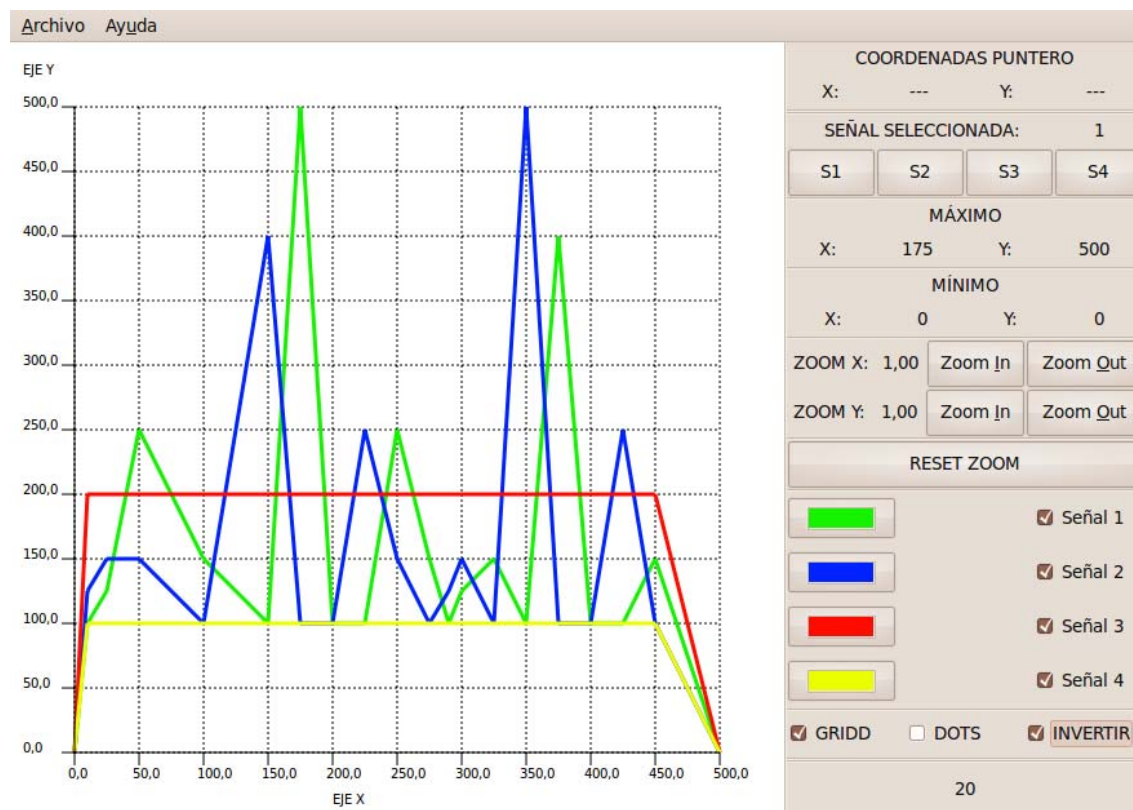


- 2) Desde el terminal insertando los datos a representar a continuación del nombre de la aplicación sin ninguna otra modificación.





En cualquier caso, el resultado de la carga de datos desde un fichero o desde el terminal sería el mismo como muestra la siguiente captura:

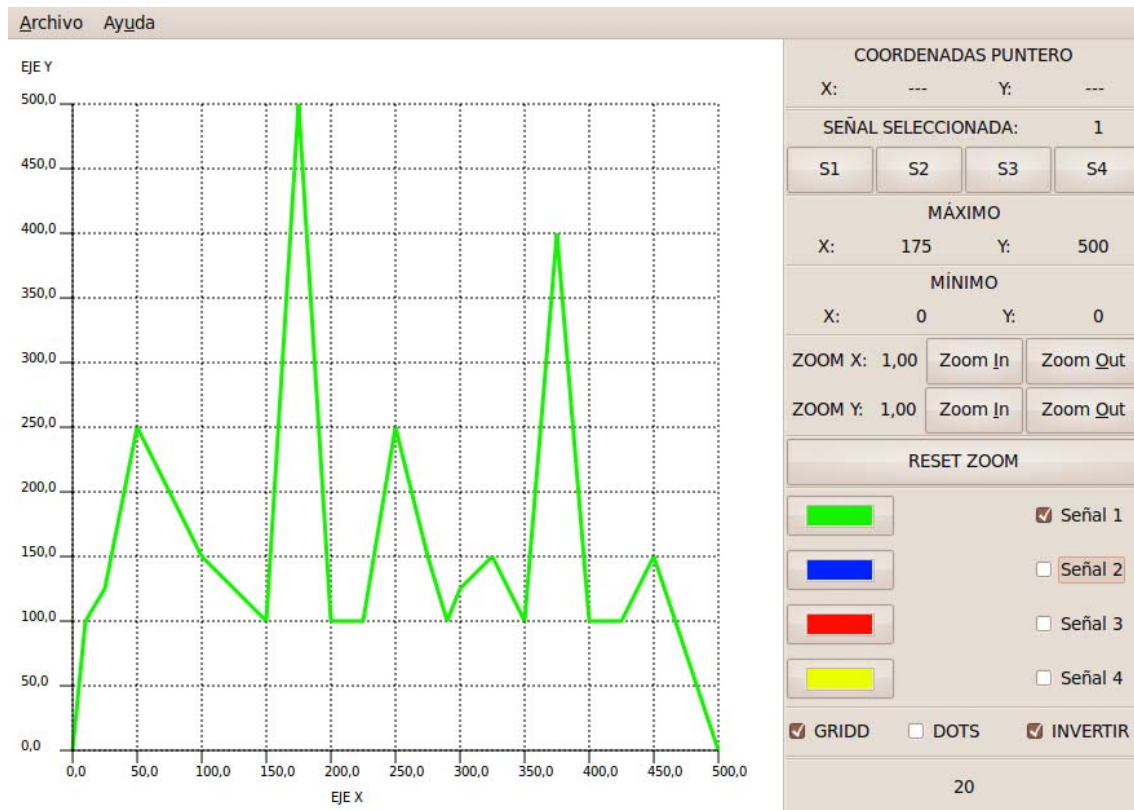




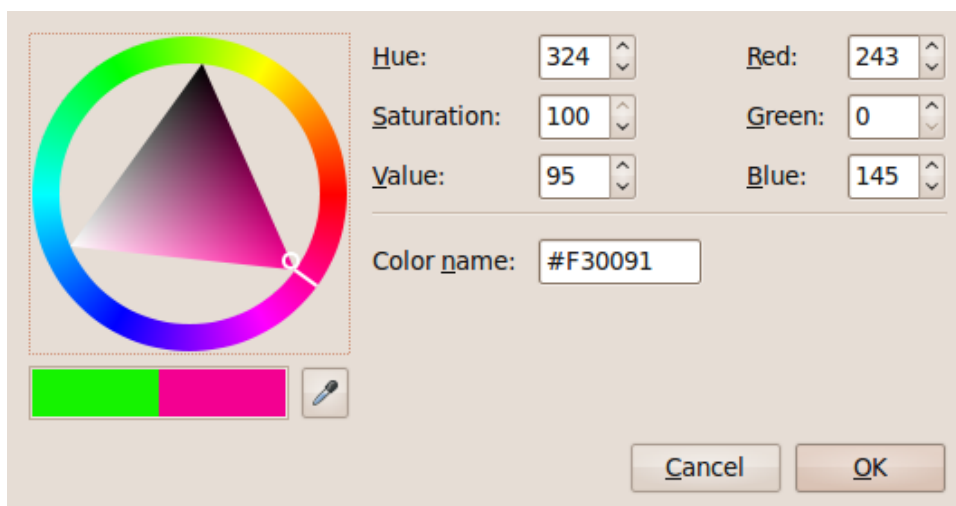


## 5.5 CAMBIAR EL COLOR DE UNA SEÑAL

Cambiar el color de cualquiera de las señales es algo trivial. En el ejemplo se muestra una señal representada en color verde:

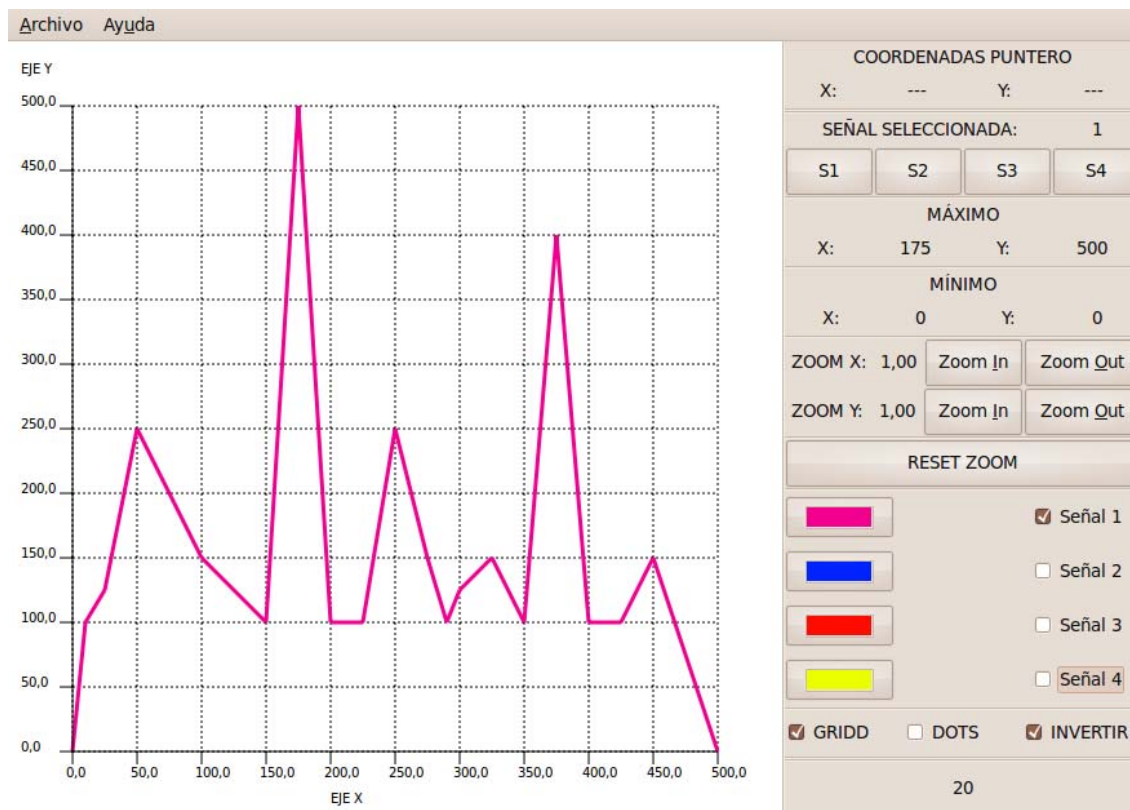


Simplemente hay que pulsar el botón de selección de color de la señal deseada para que se abra la ventana con la paleta de colores y elegir el que se desee.





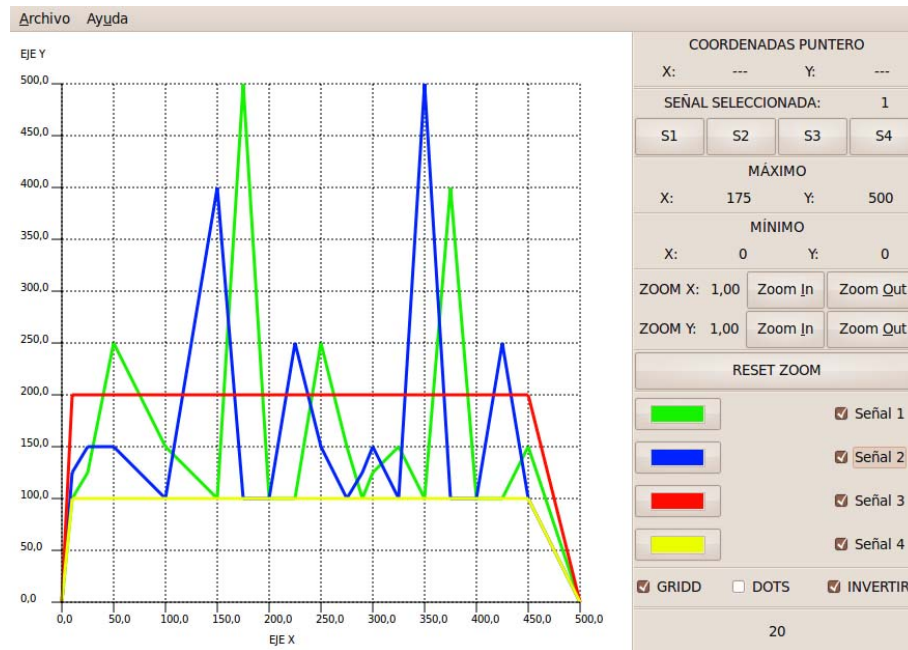
Al pulsar el botón “OK” se actualiza el color de forma instantánea.



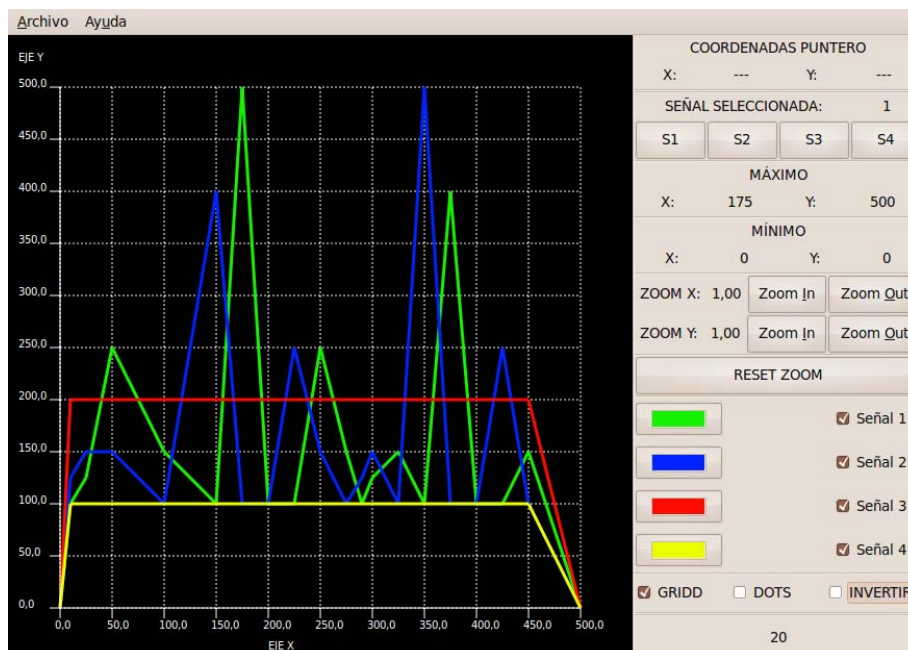
## 5.6 CAMBIAR EL COLOR DE FONDO

Para cambiar el color de fondo sólo hay que activar o desactivar la opción INVERTIR en la columna de herramientas de la derecha.

La siguiente captura muestra la visualización más adecuada para incluir el gráfico en un informe o cualquier otro medio de impresión.



La siguiente captura muestra la visualización más adecuada para visualizar el gráfico en un monitor o en un proyector.

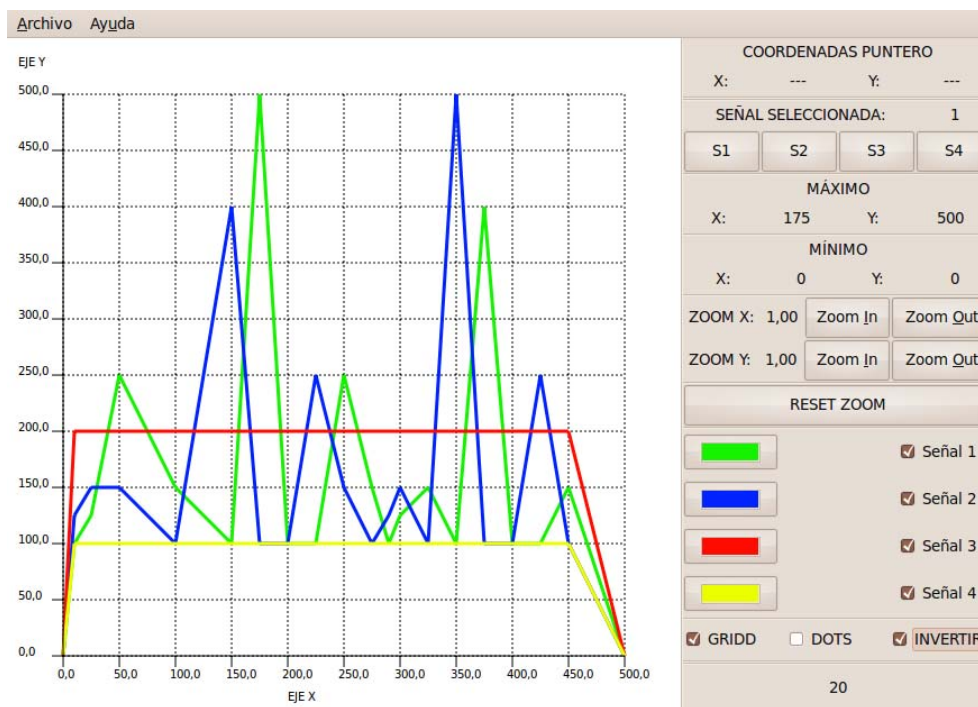




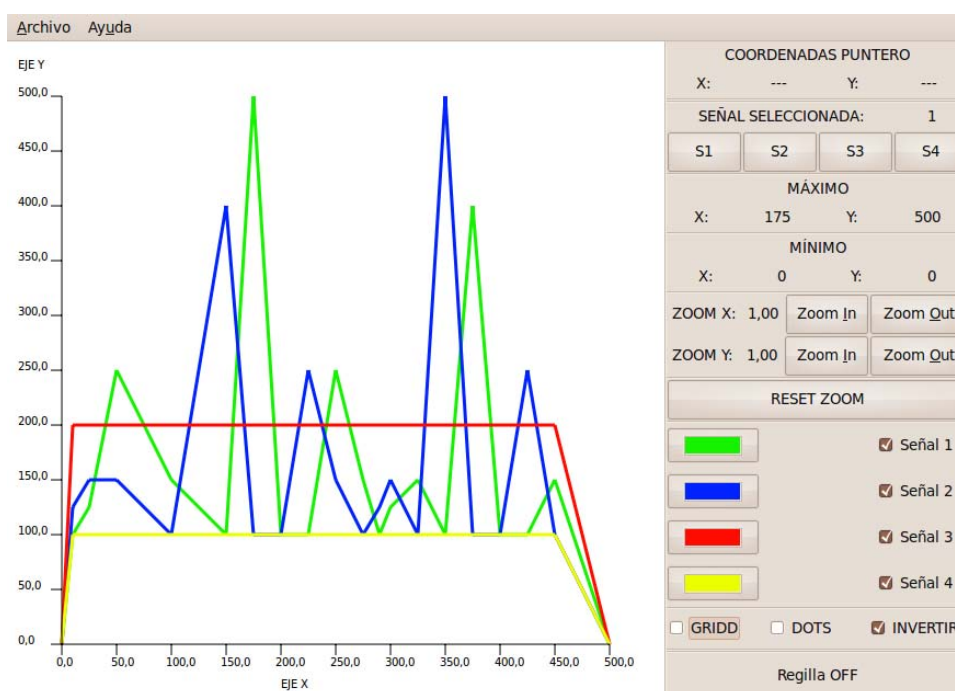
## 5.7 DESACTIVAR LA REJILLA

Para desactivar la rejilla sólo hay que desactivar la opción GRIDD en la columna de herramientas de la derecha.

La siguiente captura muestra la visualización con la opción GRIDD activada.



La siguiente captura muestra la visualización con la opción GRIDD desactivada.

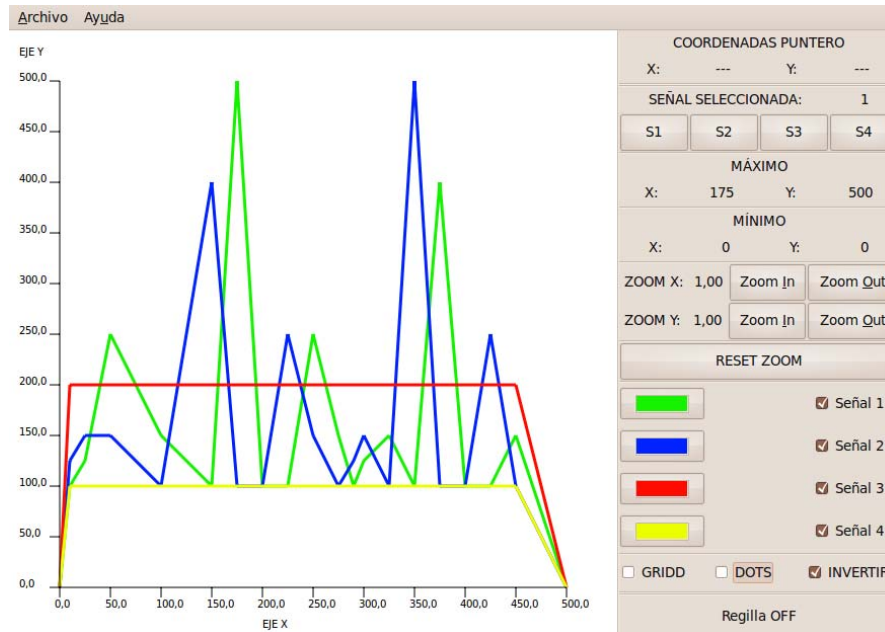




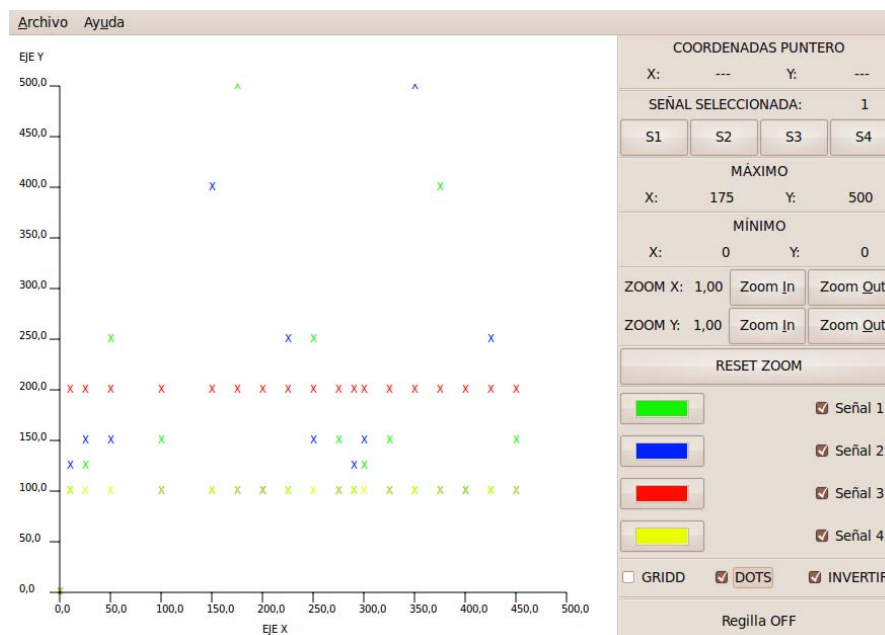
## 5.8 MOSTRAR PUNTOS O SEGMENTOS

Para mostrar los puntos sólo hay que activar la opción DOTS en la columna de herramientas de la derecha.

En la siguiente captura se muestra un ejemplo de señal con DOTS desactivado.



La siguiente captura muestra un ejemplo de señal con DOTS activado.



En ambos ejemplos la rejilla está desactivada, a propósito, para mejorar la visualización pero tienen un funcionamiento independiente.

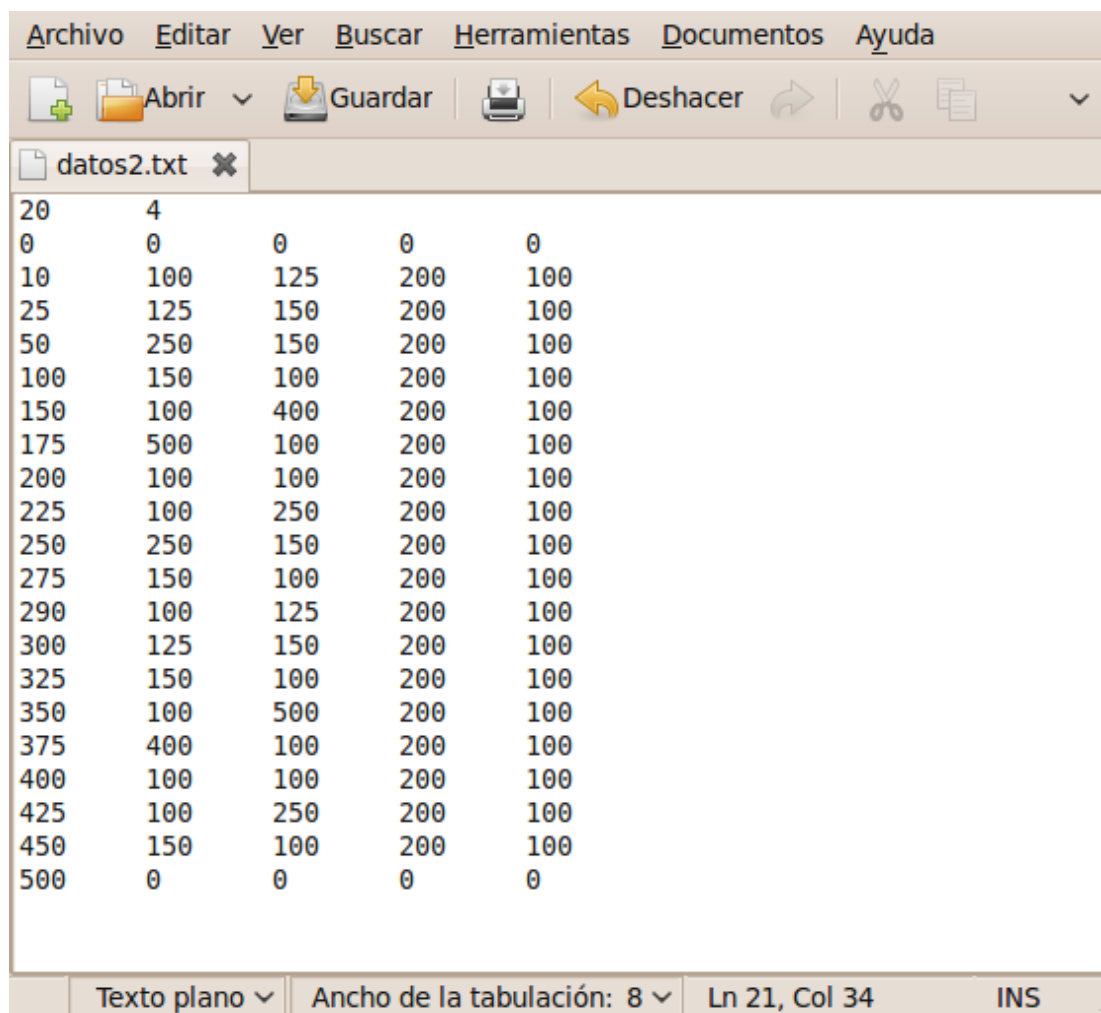




## 5.9 CÁLCULO DE MÁXIMOS Y MÍNIMOS

El cálculo de máximos y mínimos se hace de forma automática simplemente seleccionando cuál de las cuatro señales nos interesa. La aplicación está concebida de tal forma, que nos indica el primer valor máximo y mínimo, es decir, el valor de coordenada Y más alto y más bajo que aparecen por primera vez en la ventana de visualización.

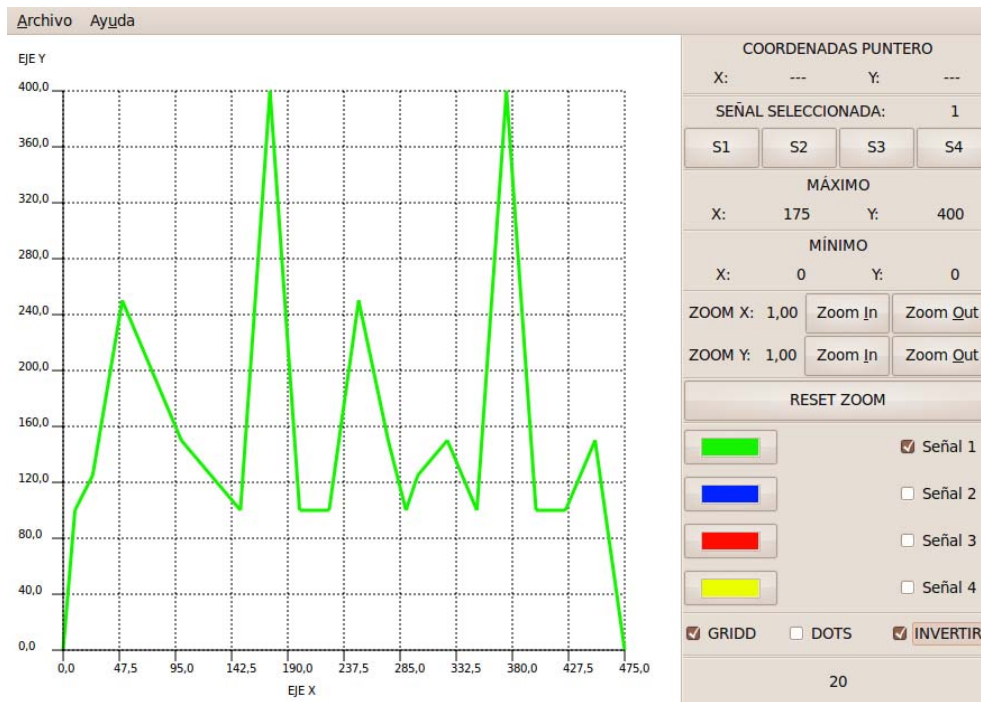
La siguiente captura indica los valores de la señal que se va a usar como ejemplo.



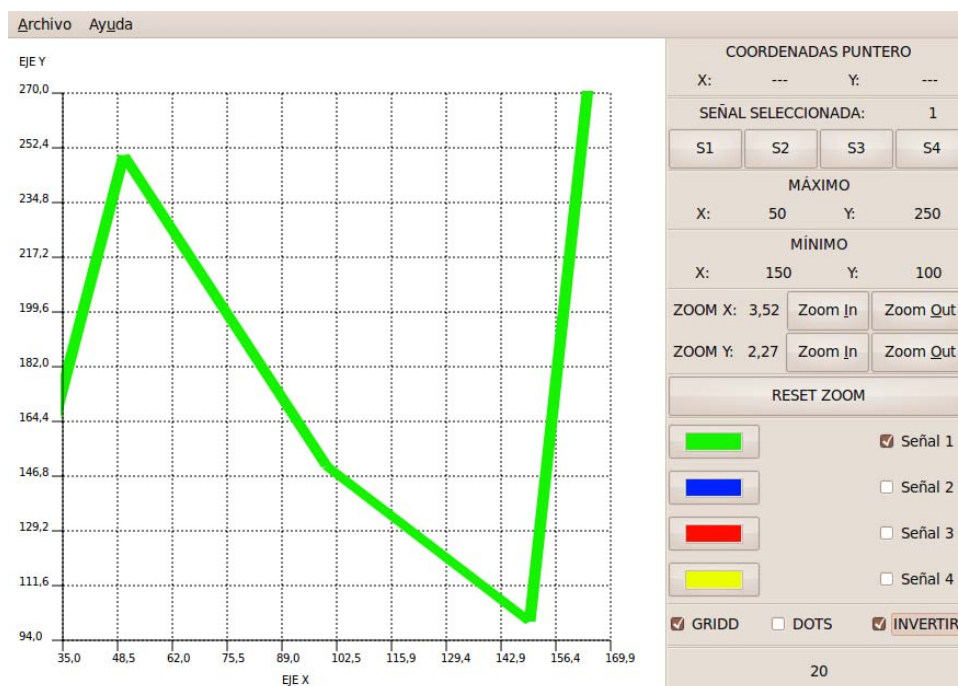
20	4			
0	0	0	0	0
10	100	125	200	100
25	125	150	200	100
50	250	150	200	100
100	150	100	200	100
150	100	400	200	100
175	500	100	200	100
200	100	100	200	100
225	100	250	200	100
250	250	150	200	100
275	150	100	200	100
290	100	125	200	100
300	125	150	200	100
325	150	100	200	100
350	100	500	200	100
375	400	100	200	100
400	100	100	200	100
425	100	250	200	100
450	150	100	200	100
500	0	0	0	0



En la siguiente captura se muestran los valores máximos y mínimos de la señal que se está visualizando. Se han desactivado el resto de señales para mejorar la visualización.



En la siguiente captura se muestra la misma señal después de ampliar una zona determinada según indica la escala. Los valores de máximos y mínimos son actualizados automáticamente.



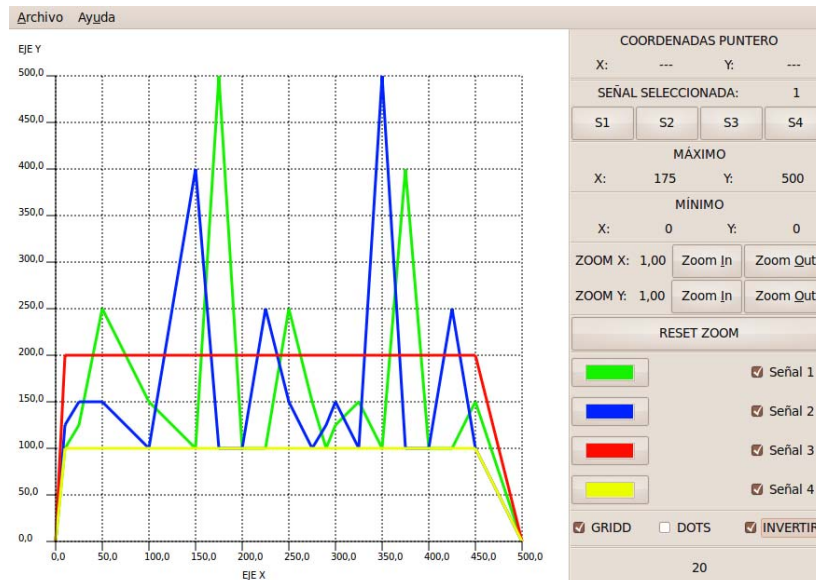


## 5.10 ZOOM

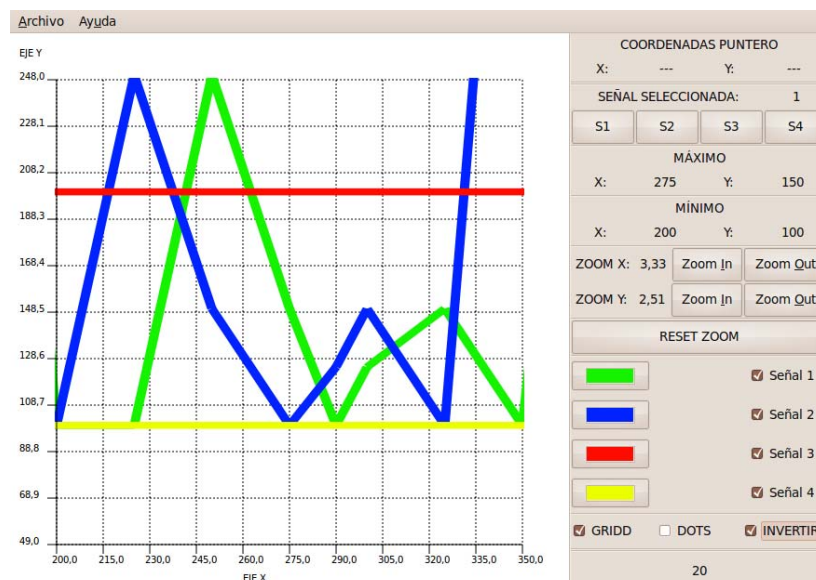
La ampliación y reducción de la señal representada en cada momento se puede hacer de dos formas distintas.

- a) Utilizando el ratón, arrastrando y soltando, se amplía el rectángulo seleccionado. Se puede realizar en cascada las veces que sea necesario. Para deshacer la última ampliación se pulsa con el botón izquierdo. Y para dejar el zoom por defecto se pulsa el botón RESET ZOOM.

La siguiente captura muestra una señal sin tener aplicado ningún zoom.



La siguiente captura muestra el resultado después de seleccionar un área de la ventana de visualización.

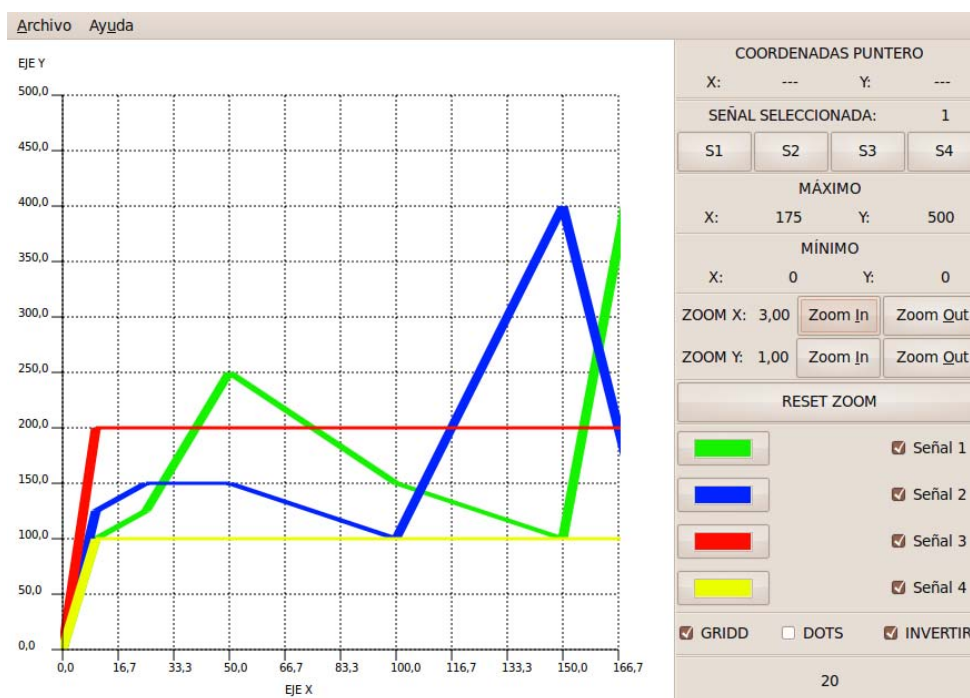




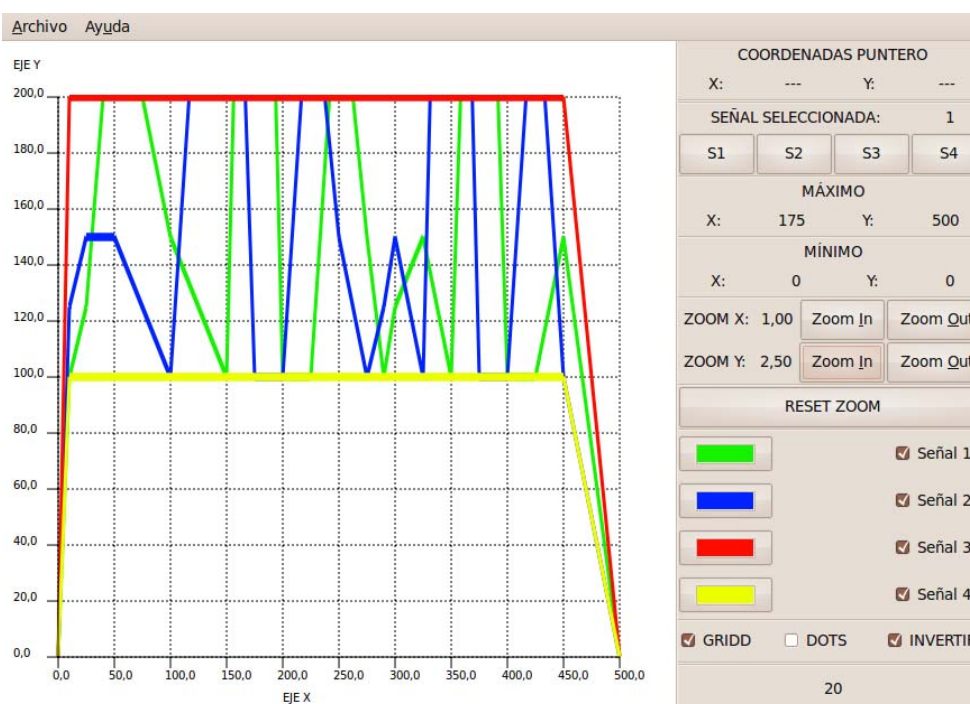


- b) También se puede ampliar la señal usando los botones disponibles en la columna de herramientas de la derecha. Esto permite modificar la proporción tanto del eje Y como del eje X con incrementos de 0.5.

En la siguiente captura se muestra el resultado de ampliar la señal en el eje X pulsando seis veces el botón.



La siguiente captura representa un ejemplo como el anterior pero en el eje Y.

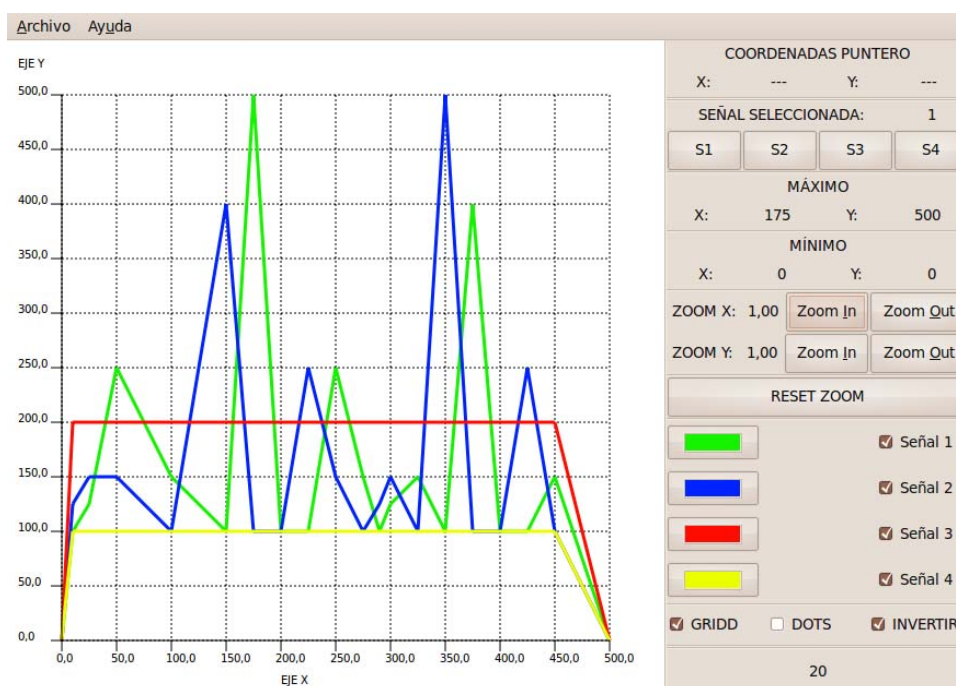




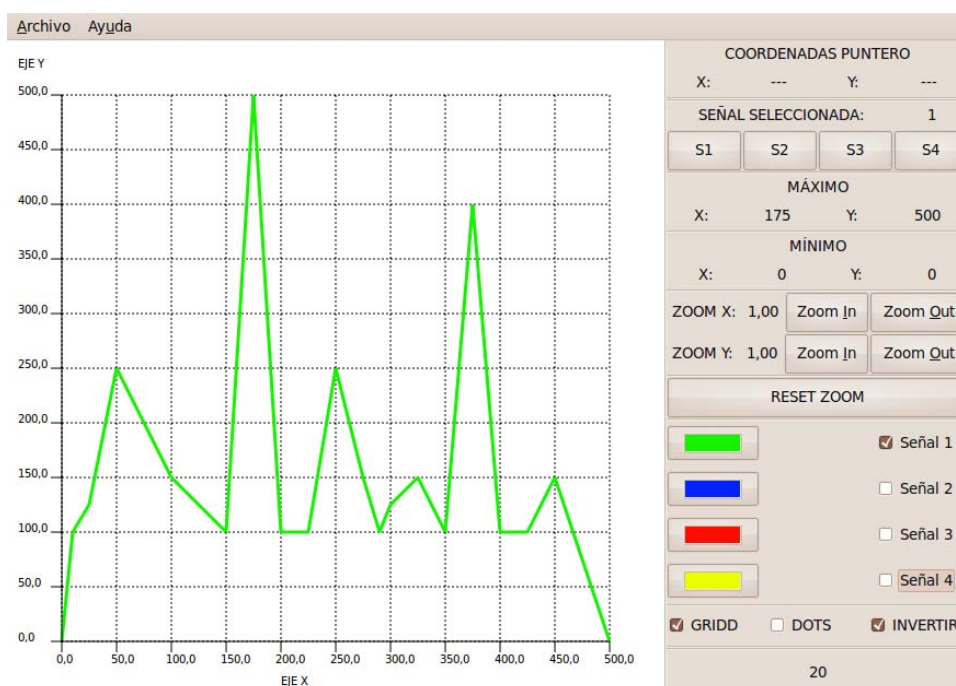
## 5.11 SELECCIÓN DE SEÑALES

En la columna de herramientas de la parte derecha de la ventana, hay 4 botones de selección para poder elegir qué señales se muestran y cuáles no. Por supuesto, sólo se muestran las señales en el caso de que existan, es decir, si el fichero de datos cargado sólo tiene dos señales de nada sirve que activemos o desactivemos las señales 3 y 4.

En la siguiente captura se presenta un ejemplo de la ventana de representación con un fichero de datos que tiene cuatro señales y se muestran simultáneamente.



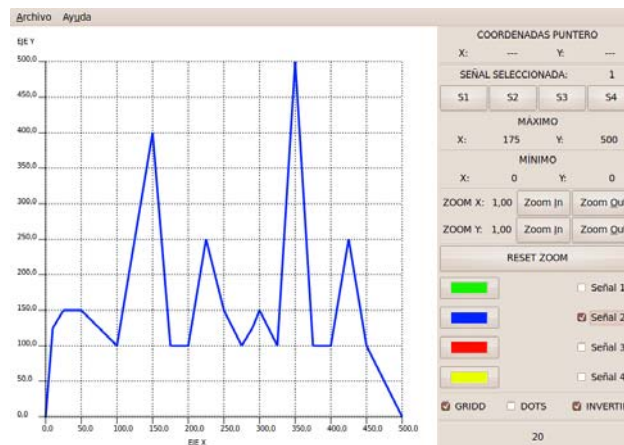
En el siguiente ejemplo se muestra la señal 1.



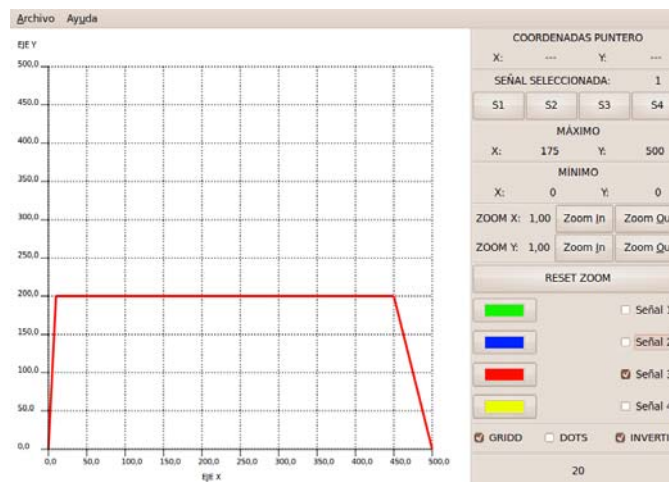


## APLICACIÓN DE REPRESENTACIÓN GRÁFICA DE DATOS Y/O SEÑALES EN ENTORNO GNU/LINUX

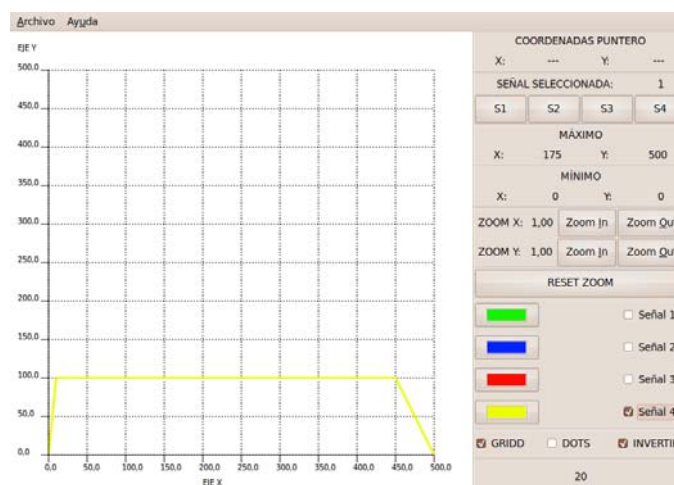
En el siguiente ejemplo se muestra la señal 2.



En el siguiente ejemplo se muestra la señal 3.



En el siguiente ejemplo se muestra la señal 4.

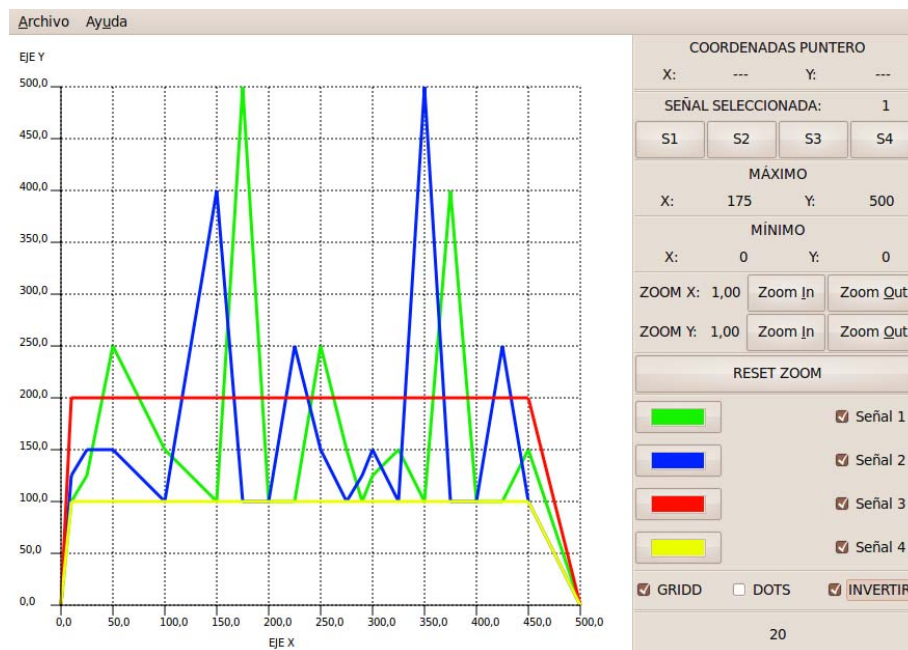


Aunque únicamente en el primer ejemplo se visualizan varias señales superpuestas, se pueden combinar de forma que se adapte a las necesidades del usuario.

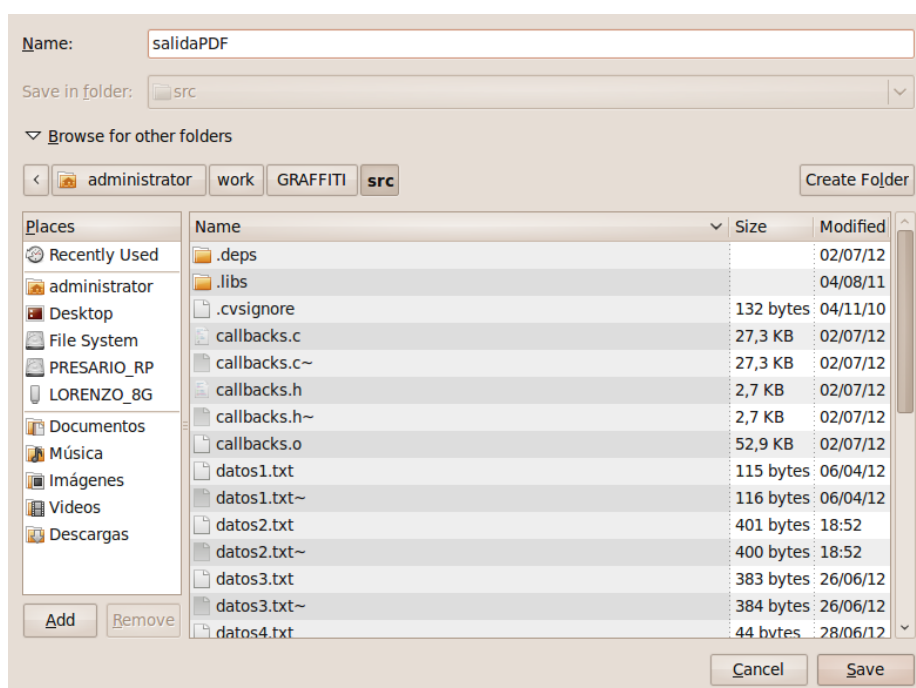


## 5.12 EXPORTAR A PDF

En el menú desplegable de la parte superior, aparece esta opción que abre una ventana para seleccionar la ruta y el nombre del fichero de salida. No es necesario indicar el tipo del fichero PDF ya que la aplicación lo hace automáticamente.



La siguiente captura muestra la ventana para seleccionar tanto la ruta como el nombre del fichero de salida, que se pueden decidir de forma arbitraria.

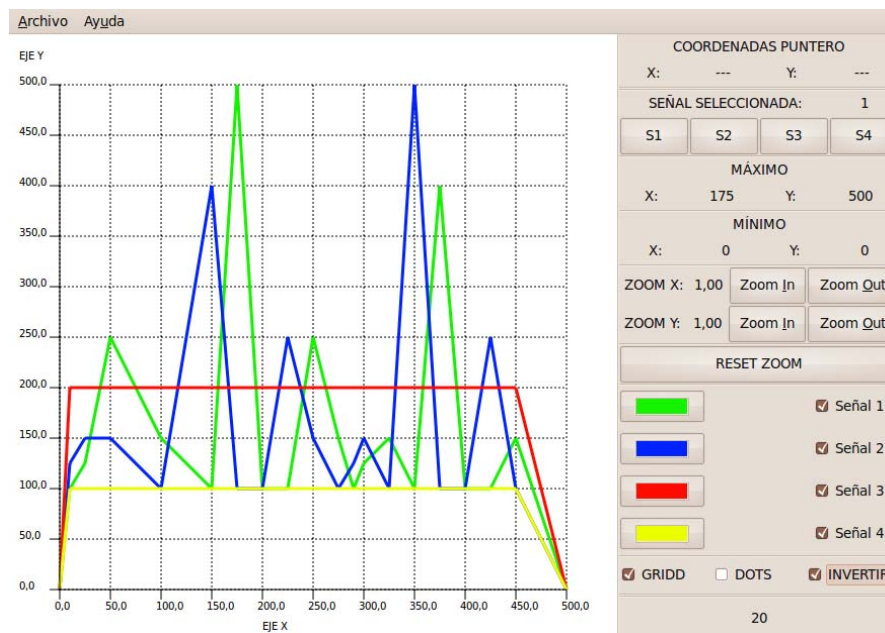




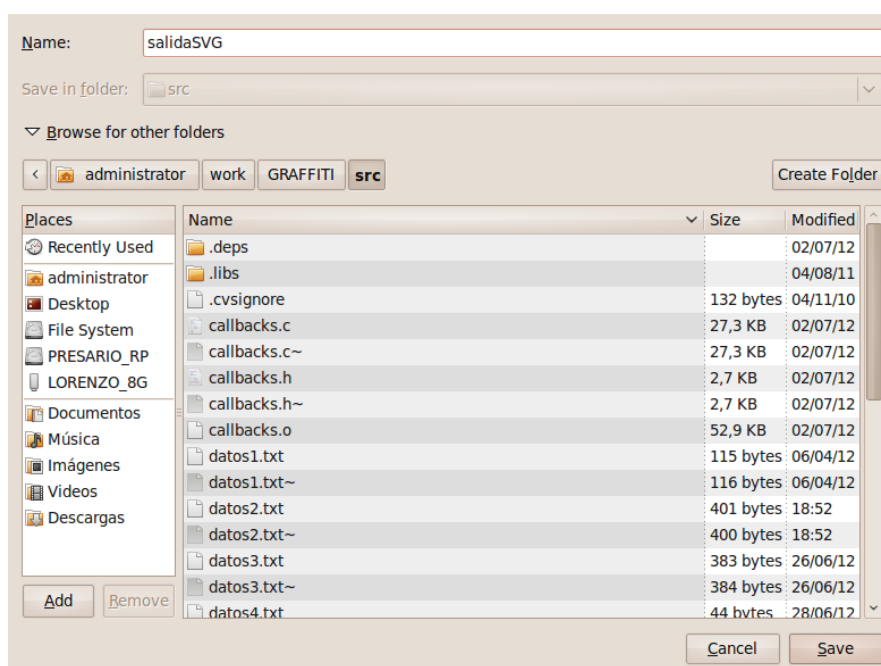


### 5.13 EXPORTAR A SVG

En el menú desplegable de la parte superior aparece esta opción que abre una ventana para seleccionar la ruta y el nombre del fichero de salida. No es necesario indicar el tipo del fichero SVG ya que la aplicación lo hace por el usuario.



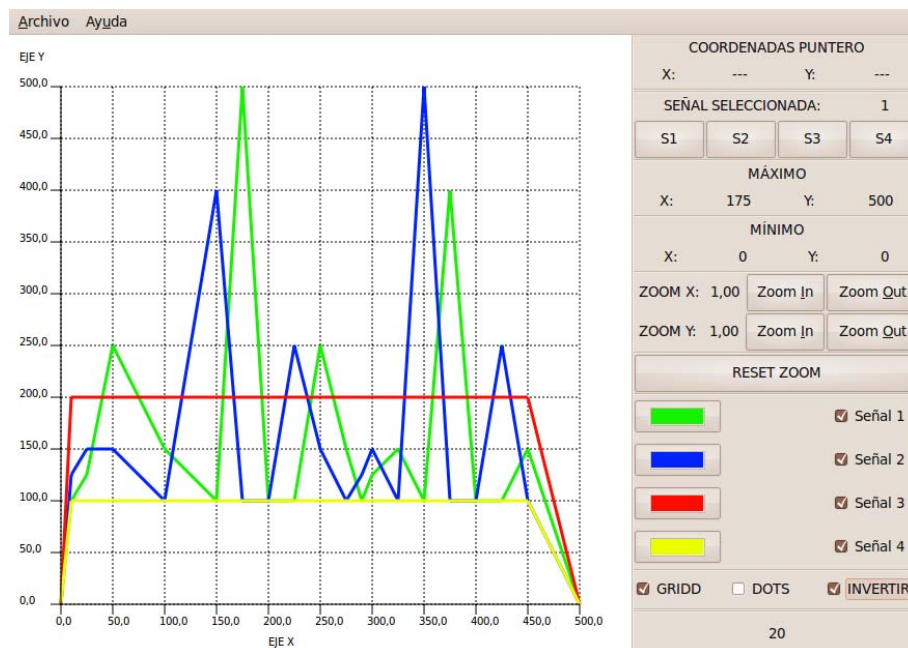
La siguiente captura muestra la ventana de selección, tanto de la ruta como del nombre del fichero de salida, que se pueden elegir de forma arbitraria.



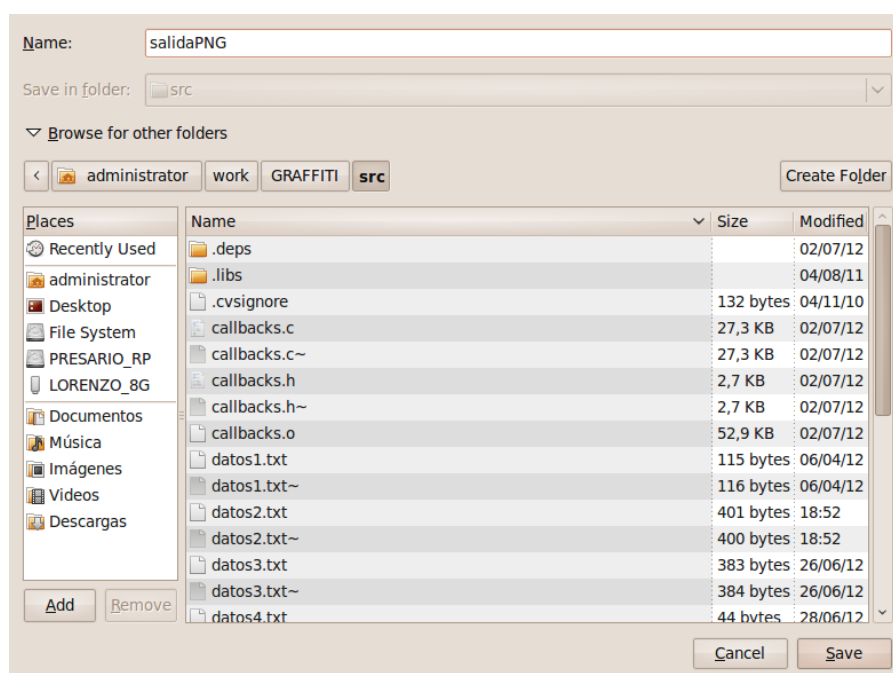


## 5.14 EXPORTAR A PNG

En el menú desplegable de la parte superior aparece esta opción que abre una ventana para seleccionar la ruta y el nombre del fichero de salida. No es necesario que se indique el tipo del fichero PNG ya que la aplicación lo hace por defecto.



La siguiente captura muestra la ventana para seleccionar, tanto la ruta como el nombre del fichero de salida, que se pueden decidir de forma arbitraria.





## **6. CÓDIGO FUENTE**

El código fuente de la aplicación no está incluido en la presente memoria por ser demasiado extenso. No obstante, se adjunta dicho código fuente dentro del CD entregado junto con la memoria.



## 7. FUTURAS LÍNEAS DE TRABAJO

A partir del diseño e implementación del presente Proyecto se podría modificar la aplicación para que se adaptara a las siguientes funcionalidades:

- Permitir elegir el idioma (especialmente Inglés) en el que aparecen todos los botones y mensajes de la aplicación.
- Integrar la aplicación en otro programa que realice captura y análisis de datos de sistemas de radio y comunicaciones.
- Calcular la frecuencia de funciones periódicas.
- Realizar la carga de datos en tiempo real (mientras se está haciendo la captura).
- Mejorar la compatibilidad, codificando la aplicación en otros lenguajes de programación y otros sistemas operativos.





## 8. CONCLUSIONES

Una vez terminado el presente Proyecto, he comprobado el potencial que tiene el desarrollo de aplicaciones GNU/LINUX. Hay una gran cantidad de documentación y de bibliotecas con funciones, que facilitan realizar múltiples tareas, especialmente en el momento de la implementación.

Además todo el entorno es software gratuito y lo que es más importante software libre, lo cual implica que el código fuente está accesible, permitiendo realizar cualquier modificación y mejora para adaptarse a las necesidades particulares de cada programador o de cada usuario final.

Para todo esto he tenido que aprender mucho sobre Ubuntu y todas sus herramientas ya que nunca había utilizado este sistema operativo, aunque sí otros similares.

Aunque poseía conocimientos previos de programación en lenguaje C, he tenido que aprender y adaptarme al entorno de desarrollo y a las peculiaridades del sistema, una tarea dura que me ha llevado algún tiempo, pero he afianzado mi destreza programando.

He aprendido a realizar todos los pasos del desarrollo software, desde el análisis de los requisitos, hasta el diseño y la implementación. Esto me ha llevado a comprobar la importancia de una buena metodología de trabajo para conseguir unos resultados satisfactorios, aplicando así, todos los conocimientos adquiridos en mi formación universitaria.

En cuanto a la documentación, a pesar de disponer de mucha en español y en inglés, ha sido laboriosa la búsqueda de algunos temas, por el carácter específicamente gráfico de esta aplicación.

Con este proyecto he alcanzado mayor experiencia a la hora de analizar, diseñar y codificar aplicaciones y en muchos momentos “a buscarme la vida” con el fin de cumplir las especificaciones.



El resultado ha sido una aplicación que cumple adecuadamente las funcionalidades requeridas y que bien usada directamente o realizando las modificaciones pertinentes, permitirá a muchos programadores y usuarios el desarrollo de aplicaciones o sistemas de adquisición de datos más complejos, sin tener que preocuparse de la representación de señales en pantalla y poder analizar dichas señales con más detalles gracias a un zoom fácil de usar.

Además, con la opción de exportar a distintos formatos, facilita la operación de realizar informes o documentar pruebas o experimentos sin tener que recurrir a repetidas capturas de pantalla en las que se pierde definición de la imagen.



## 9. BIBLIOGRAFÍA

La bibliografía que se ha utilizado en el presente Proyecto básicamente consiste en numerosas páginas WEB en las que se han consultado desde pequeños manuales y tutoriales a foros de programación con ejemplos de código.

<http://www.ubuntu-es.org/>

<http://www.gnu.org/gnu/linux-and-gnu.es.html>

<http://www.estrellateyarde.org/discover/manual-c-anjuta-en-linux>

<http://www.gnu.org/home.es.html>

<http://www.anjuta.org/>

<http://glade.gnome.org/>

<http://www.gtk.org/>

<http://cairographics.org/>

<http://www.ayuda-internet.net/tutoriales/manu-unix/manu-unix.html>

<http://www.gnu.org/philosophy/free-sw.es.html>

<http://www.debian.org/index.es.html>

<http://developer.gnome.org/glib/>

<http://www.es.gnome.org/>

<http://www.masadelante.com/faqs/pdf>



<http://www.alegsa.com.ar/Dic/svg.php>

<http://www.w3.org/Graphics/SVG/>

<http://es.scribd.com/doc/454427/25/Modelo-basado-en-prototipado-incremental>



## 10. GLOSARIO

**ANJUTA:** Es un entorno de desarrollo integrado versátil con una serie de facilidades de programación avanzadas, que incluye la gestión de proyectos, asistente de la aplicación, un depurador interactivo, editor de código fuente, el control de versiones, diseñador de interfaz gráfica de usuario, perfiles y muchas herramientas más.

**CAIRO:** El Cairo es una biblioteca de gráficos en 2D con soporte para múltiples dispositivos de salida.

**DEBIAN:** Es un sistema operativo libre que utiliza el núcleo Linux, pero la mayor parte de las herramientas básicas vienen del Proyecto GNU/LINUX.

**GDK (GIMP Drawing Kit):** Es una biblioteca de gráficos que actúa como un intermediario entre gráficos de bajo nivel y gráficos de alto nivel.

**GLADE:** Es una aplicación para permitir el desarrollo rápido y fácil de interfaces de usuario para GTK y el entorno de escritorio GNOME.

**GLIB:** Biblioteca que proporciona los bloques básicos para construir aplicaciones. Suministra el conjunto de objetos básicos usado en GNOME, la implementación del bucle principal, y un gran conjunto de funciones de utilidad para cadenas y estructuras de datos comunes.

**GNOME:** Es un escritorio fácil de utilizar para sistemas GNU/Linux o UNIX.

**GNU:** Es un proyecto de sistema operativo completo tipo Unix de software libre.

**GTK (GIMP Drawing Kit):** Es una herramienta multiplataforma para crear interfaces gráficas de usuario. GTK es adecuado para proyectos de cualquier envergadura y hace uso de las bibliotecas de GDK.



**IDE (Integrated Development Environment):** Un entorno de desarrollo integrado es un programa informático compuesto de herramientas de programación (editor, compilador, depurador, etc.) Además puede soportar uno o varios lenguajes de programación.

**LINUX:** Es un sistema operativo, compatible con Unix. Las dos principales características que lo definen son que es libre y que el sistema viene acompañado del código fuente lo que permite cualquier tipo de modificación y cambio.

**PDF (Portable Document Format):** Es el formato de archivos desarrollado por Adobe Systems. Esta tecnología ha tenido éxito estandarizando el formato de los documentos que se utilizan y transfieren en Internet. El PDF es como un formato de archivos universal.

**PNG (Portable Network Graphics):** Es un formato gráfico no sujeto a patentes basado en un algoritmo de compresión sin pérdidas para bitmaps. Este formato fue desarrollado para solventar las deficiencias del formato GIF y permite almacenar imágenes con una mayor profundidad de contraste y otros importantes datos.

**SOFTWARE LIBRE:** Software que se basa en el concepto de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

**SVG (Scalable Vector Graphics):** Lenguaje abierto que permite crear gráficos vectoriales 2D basado en XML, tanto estáticos como animados. La gran popularidad de este formato, se debe a que no tiene costes de patentes su utilización y la ventaja que supone el no perder calidad si una imagen en este formato es ampliada.

**UBUNTU:** Es una distribución GNU/LINUX que ofrece un sistema operativo principalmente enfocado a ordenadores de escritorio, aunque también proporciona soporte para servidores.



**UNIX:** Es un sistema operativo que tiene su origen en los laboratorios Bell de AT&T en los años 60. Es la base de numerosos sistemas operativos actuales, como por ejemplo LINUX.

**WIDGETS:** En este contexto se refiere a las diferentes partes de una interfaz de usuario, por ejemplo botones, etiquetas, cuadros de texto, área de dibujo, etc.